



**Technická univerzita  
v Liberci**

---

Fakulta mechatroniky a mezioborových  
inženýrských studií

# **DIPLOMOVÁ PRÁCE**



# Technická univerzita v Liberci

---

Fakulta mechatroniky a mezioborových  
inženýrských studií

Studijní program: 2612M – Elektronika a informatika

## Sofistikované metody pro zpracování obrazu The sophisticated methods of image processing

Karel Kněžourek

Vedoucí DP : RNDr. Aleš Linka, CSc.

Počet stran:	57
Počet obrázků:	15
Počet tabulek:	11
Počet příloh:	3
Počet pramenů:	10

## **Anotace**

KNĚŽOUREK, Karel 2003

Ved. DP: RNDr. Aleš Linka, CSc.

### **Sofistikované metody pro zpracování obrazu**

Cílem této práce je vytvoření internetové aplikace, pomocí níž bude možné na obrazové soubory aplikovat postupy pro zpracování a analýzu obrazu. Aplikace obsahuje postupy jak nově vyvinuté, tak i všeobecně používané. Jedná se o postupy : filtrace obrazu, transformace obrazu a segmentace obrazu. Vhodnými prostředky pro programování aplikace byly Apache Web Server s podporou PHP a Matlab s komunikačním toolboxem. V poslední kapitole jsou demonstrovány tyto metody na ukázkovém obrazovém souboru.

## **Abstract**

KNĚŽOUREK, Karel 2003

Supervisor DT: RNDr. Aleš Linka, CSc.

### **The sophisticated methods of image processing**

The aim of this work is create web's program by reason to apply the technique for processing and analysis image on choosen image file. The web's program contain procedures as newly developed so generally used. Digital image processing methods are considered: filtering image, transforming image and segmenting image. Advisable application for programming this web's program are Apache Web Server with the support communications toolbox. In conclusion are demonstrated the methods on exemplary image file.

## **Prohlášení**

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména §60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o využití mé DP a prohlašuji, že souhlasím s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

# Obsah

1.	Úvod	8
2.	Nastavení serveru	9
2.1.	Výběr programovacích prostředků	9
2.2.	WWW server	10
2.2.1.	Instalace Apache pod systémem Windows	10
2.2.2.	Konfigurace WWW Serveru Apache	11
2.2.3.	Spouštění Apache v prostředí MS Windows	18
2.3.	PHP	19
2.3.1.	PHP instalace	19
2.4.	Matlab	20
2.4.1.	Matlab Web Server	21
3.	Popis aplikace pro zpracování obrazu	22
3.1.	Popis způsobu komunikace	22
3.2.	Hlavní Matlab funkce	23
3.3.	Ovládání aplikace	24
3.3.1.	Spouštění aplikace	24
3.3.2.	Popis rozložení	25
3.3.3.	Položky v menu	26
3.3.4.	Logování činnosti	31
3.3.5.	Omezení aplikace	31
3.3.6.	Možnost rozšiřování aplikace	32
3.3.7.	Ukázky aplikace	34

4.	Řešení algoritmů v Matlabu	36
4.1.	Předmluva	36
4.2.	Zpracování obrazu	36
4.3.	Použité funkce Image Processing Toolboxu	36
4.4.	Vytvořené funkce	37
4.5.	Využívání DLL knihoven	40
4.6.	Problémy při torbě Matlab funkcí	41
4.7.	Postupy	42
5.	Ukázky	43
5.1.	Ukázka 1	43
5.2.	Ukázka 2	45
5.3.	Ukázka 3	46
6.	Závěr	52
	Literatura	53
	Přílohy	54
	Přílohy A	54
	Přílohy B	56
	Přílohy C	57

## Zkratky

<b>Apache</b>	Apache HTTP SERVER PROJECT. Volně dostupný a nejrozšířenější WWW server.
<b>CGI</b>	Common Gateway Interface.
<b>CSS</b>	Cascading Style Sheets. Kolekce metod pro grafickou úpravu stránek
<b>MWS</b>	Matab Web Server.
<b>HTTP</b>	HyperText Transfer Protocol. Protokol používaný službou WWW sítě Internet.
<b>m-file</b>	Skript, nebo funkce Matlabu.
<b>PHP</b>	Hypertextový preprocesor, nástroj pro generování dynamických WWW stránek.
<b>Java</b>	Síťově orientovaný interpretovaný programovací jazyk
<b>Applet</b>	Java aplikace spustitelná v WWW prohlížeči.
<b>JavaSkript</b>	Odkaz jazyka Java, jeho interpret je obsažen ve většině prohlížečů
<b>MIME</b>	Multipurpose Internet Mail Extension. Standart pro identifikaci a kódování souborů, které jsou posílány přes internet.
<b>DNS</b>	Domain Name System (Server). DNS server zná síťové adresy všech počítačů v síti. Převádí symbolická jména na IP adresy, vytváří v síti hierarchické, nezávisle spravované domény.
<b>DLL</b>	Knihovna funkcí systému Windows.

# 1. Úvod

Nezadržitelný rozvoj internetu s sebou přináší nové možnosti programování internetových aplikací. Potřebná vstupní data jsou získávána od uživatele z internetového prohlížeče. Po zpracování dat jsou výsledky výpočtů vráceny zpět uživateli ve formě HTML kódu. K prezentaci výsledků můžeme použít i nevykonný počítač, který vyžaduje jen internetový prohlížeč.

Cílem této práce je vytvoření internetové aplikace pro zpracování obrazu, která bude využívat Matlab Web Server. V rámci této aplikace budou rovněž vytvořeny vhodné funkce v Matlabu nebo Java applety pro zpracování obrazu.

Matlab Web Server je nástroj, který zpřístupňuje některé výpočetní možnosti Matlabu přes HTTP protokol. Tento server je schopen přijmout data zaslaná prohlížečem, spustit obslužný program a předat mu tato data jako vstupní parametry. Výsledky těchto obslužných programů pak mohou být prezentovány nejen v textové podobě, ale i graficky.

Vzhledem k velkému přenosu informací bylo navíc nezbytné snížit redundantní data. Dále bylo nutné zvýšit srozumitelnost vstupních a výstupních dat pro další možné rozšiřování aplikace. Snížením velikosti přenášených dat získáme rychlejší běh aplikace. K naprogramování tohoto řízení je využit programovací jazyk PHP.

Pro správný běh aplikace je nutné mít správně nainstalován WWW server s podporou CGI. Pro vlastní realizaci byl použit WWW server APACHE.

Implementace algoritmů a postupů pro analýzu obrazu s ohledem na tvorbu plnohodnotné webové aplikace je natolik obsáhlý problém, že cílem není implementace většiny dostupných algoritmů, ale spíše vybrat takové postupy, které lze efektivně používat v rámci webové aplikace nebo postupy, které lze sofistikovaně upravit tak, aby tuto podmínku splňovaly. Na tyto algoritmy a postupy jsou především kladeny zvýšené nároky z hlediska rychlosti výpočtů a velikosti výstupů.

Vlastní práce je strukturována do následujících kapitol. Správné nastavení WWW serveru Apache, PHP a MWS je popsáno v první kapitole. Použité algoritmy a postupy jsou popsány v kapitole 2. Poslední kapitola se demonstruje výsledky těchto algoritmů. V příloze A jsou ukázky kódů PHP, matlab funkcí a Javy. Požadavky na hardware a software najdeme v příloze B. Součástí této práce je CD ROM se zdrojovými kódy, jehož obsah je uveden v příloze C.



## 2. Nastavení serveru

Algoritmy pro zpracování obrazu mají obecně vysoké požadavky na hardware počítače. Pokud je počítačem internetový server jsou tyto nároky daleko větší. Špatně nastavený server vede k jeho zatěžování a může způsobit zhroucení celého systému. V této kapitole je popsáno správné nastavení serveru.

### 2.1. Výběr software pro server

Vhodným výběrem software pro server lze zvýšit jeho výkon a tím i rychlost internetových aplikací, které jej budou využívat. Důvody, které vedly k vybrání níže popsaného software uvádím v následujících bodech:

- Budeme přenášet velké objemy dat a to zejména obrazových. Z tohoto důvodu je nutné mít minimální velikost zdrojových kódů a obrazové data ukládat v minimální velikosti s ohledem na kvalitu obrazu. Na základě toho byl zamítnut např. *Macromedia Flash*.
- Snaha o platformovou univerzálnost, tj. uživatel nemusí zjišťovat jaký operační systém je nainstalován, či jaká verze a druh WWW prohlížeče se v počítači nachází.
- Možnost použití výpočetních možností Matlabu.
- V neposlední řadě jsem také uvažoval o dosažitelnosti manuálů, nápověd a finanční dostupnosti .

Zvolený software je WWW server Apache s podporou programovacího jazyka PHP, Matlab s komunikačním toolboxem. Dále jsou používány DHTML, JavaSkript a CSS styly. Při vlastní tvorbě aplikace byl rovněž použit Java applet, který sice není podporován všemi prohlížeči, ale jeho použití se ukázalo potřebné zejména v situaci, kde vývojové prostředky PHP nestačily.

## 2.2. WWW server Apache

Jedinou podmínku kladenou na WWW server je podpora CGI. Tuto podmínku vyžaduje Matlab Web Server a PHP (pokud je PHP spouštěno jako interpret CGI). WWW server Apache je v současnosti nejrozšířenější WWW server, který je možné provozovat na různých platformách operačních systémů. Díky jednoduché konfiguraci a malým nárokům na operační systém se Apache provozuje nejen na serverech s operačními systémy Unix Linux, ale i na počítačích s Microsoft Windows, Novel NetWare, BeOS, Mac, OS/2, hp Open VMS aj.

WWW server Apache byl vyvinut na základě úspěšného WWW serveru NCSA. V roce 1995 vznikla skupina vývojářů *Apache Group*, která se později změnila na *Apache Software Foundation*

<http://www.apache.org/>

Ve stejném roce přišla na svět první verze Apache. Od té doby prošel Apache úspěšným vývojem. Dnes je distribuována zatím poslední verze Apache 2.0.45.

### 2.2.1. Instalace Apache pod systémem MS Windows

Pro provozování WWW serveru Apache je doporučen operační systém MS Windows NT 4.0, respektive MS Windows 2000 či Windows XP. Pokud se rozhodnete provozovat Apache v Microsoft Windows 95, musíte aktualizovat knihovnu *Windows Socket* (WinSock) na verzi 2. V případě instalace Apache pomocí *.msi* souborů, musíte navíc v Microsoft Windows 95, 98 a NT nainstalovat aplikaci *Microsoft Windows Installer*. Odkazy na požadované programové vybavení naleznete na stránkách s binární distribucí Apache.

Existují dva různé způsoby instalace.

#### a) Kompilace zdrojového kódu

Je opět podmíněna vhodným překladačem. Doporučeným překladačem je *Microsoft Visual C++ 5.0* a vyšší.

## b) Binární instalace

V případě operačního systému Microsoft Windows se používá nejčastěji. Instalační soubory jsou k dispozici na

<http://www.apache.org/dist/httpd/binaries/win32/>

a jsou určeny pouze pro počítače s procesory Intel. Pro verzi Apache 2.0 jsou k dispozici jak soubory *.msi*, tedy pro instalaci pomocí aplikace *Microsoft Windows Installer*, tak samospustitelné v instalačních souborech *.exe*. Stejná situace je i v případě starší verze Apache 1.3. Instalace Apache v Microsoft Windows je jednoduchá. Po stažení a spuštění instalačního souboru, například *apache\_2.0.45-win32-x86-no\_ssl.msi*, se zobrazí úvodní okno průvodce instalací. Ten bude postupně zobrazovat další okna. Jako první se zobrazí s licenčním ujednáním a s popisem produktu. Následuje okno s formulářem, ve kterém vyplníte jméno domény vašeho počítače, jméno počítače a e-mailovou adresu správce WWW serveru a zvolíte, zda chcete Apache spouštět jako službu (pouze ve Windows NT a 2000) nebo manuálně. Dalším krokem je pak volba kompletní nebo volitelné instalace. Implicitním adresářem pro instalaci je *C:\Program Files\Apache Group\Apache2*. Instalační proces je ukončen zkopírováním souborů do cílového adresáře a registrací software v operačním systému MS Windows.

### 2.2.2. Konfigurace WWW Serveru Apache

#### • Adresáře Apache

Adresář s instalací Apache je rozdělen do několika podadresářů. Nejdůležitější jsou:

- bin.....Obsahuje programy pro běh WWW serveru Apache.
- htdocs.....Kořenový adresář s publikovatelnými dokumenty.
- conf.....Adresář s konfiguračními soubory.
- logs..... Adresář se soubory, které obsahují informace o činnosti  
WWW serveru.
- cgi-bin.....Adresář pro externí spustitelné soubory (CGI skripty).
- modules.....Adresář s moduly, které lze aktivovat při startu  
WWW serveru.

- **Konfigurační soubory**

Apache verze 2.0 používá tyto konfigurační soubory:

- `httpd.conf`.....Hlavní konfigurační soubor serveru (základní nastavení).
- `mime.types`..... Popis podporovaných typů dokumentů (MIME typ) a jim odpovídajících přípon souborů.
- `magic`.....Soubor, který obsahuje vzorky dokumentů. Tyto vzorky se použijí pro rozpoznání typu souboru v případě, kdy nelze podle jinak zjistit MIME typ souboru.
- `.htaccess`.....Tento konfigurační soubor je uložen v jakémkoli adresáři, který obsahuje publikovatelné dokumenty. Nastavuje vlastnosti adresáře. Jeho direktivy, které jsou totožné s direktivami v `httpd.conf`. Pokud je použití `.htaccess` povoleno, je upřednostňován před stejnými direktivami v hlavním konfiguračním souboru.

Starší verze Apache 1.2 a 1.3 navíc využívaly pro konfiguraci tyto soubory:

- `access.conf`.....Nastavení oprávnění pro servery, uživatele, CGI skripty, dokumenty atd. Nyní je součástí hlavního konfiguračního souboru `httpd.conf`.
- `srm.conf`.....Soubor popisující obsah serveru. Nyní je součástí hlavního konfiguračního souboru `httpd.conf`.

- **Konfigurační soubor `httpd.conf`**

Konfigurace Apache je stejná pro všechny platformy operačních systémů. Spočívá v editaci konfiguračního souboru `httpd.conf` adresáři `conf`, který je umístěn v instalačním adresáři Apache. Konfigurační soubor `httpd.conf` obsahuje direktivy, které se zapisují na samostatné řádky. Pro komentář nebo pro vypnutí direktivy se používá znak `#`. Soubor `httpd.conf` je rozdělen do tří částí. V první se nastavují vlastnosti serveru, ve druhé je hlavní část konfigurace a ve třetí se nastavují virtuální hostitelé. Tyto sekce slouží pro zpřehlednění a jednotlivé direktivy nemusí být vázány na tyto sekce.

## **Direktivy určené pro konfigurace serveru**

- ServerRoot.....Kořenový adresář instalace Apache.
- ServerName.....Jméno WWW serveru.
- Listen.....Komunikační port protokolu TCP/IP, na kterém Apache naslouchá požadavky od prohlížečů. Implicitní hodnota je 80. V případě nastavení virtuálních hostitelů lze nastavit více těchto direktiv. Pro rozlišení se přidá IP adresa.
- PidFile.....Souboru, který obsahuje jednoznačnou identifikaci procesu pache (PID). Tato hodnota slouží v systému Unix a Linux pro zastavení či opětovné nastartování serveru.
- ServerAdmin.....Adresa elektronické pošty správce serveru.
- LoadModule .....Aktivace DSO modulu.
- ErrorLog .....Jméno logovacího souboru s hlášením o chybách WWW serveru. Implicitně je umístěn v adresáři logs.
- CustomLog ..... Jméno správcem nastaveného logovacího souboru. Formát zpráv ukládaných do tohoto souboru se nastaví v direktivě LogFormat. Implicitně je umístěn v adresáři logs.
- LogFormat.....Formát ukládaných zpráv do logovacích souborů.
- ErrorDocument.....Jméno soubor s chybovým textem pro chybový kód.

## **Direktivy určené pro publikování dokumentů**

- DocumentRoot..... Specifikuje vrchol stromu dokumentů
- DirectoryIndex..... Jméno výchozího souboru v adresáři s dokumenty. Při zadání URL adresy, která bude odkazovat pouze na tento adresář, se tento soubor automaticky zobrazí. Standardně se používají soubory index.html.
- <Directory>.....Nastavení vlastností fyzického adresáře. Obsahuje direktivy, které se vztahují k danému fyzickému adresáři.
- <Location>.....Obdoba <Directory>, není však vztažena na konkrétní adresář, ale na URL adresu.
- <Files>..... Nastavení vlastností souborů.

- Options.....Nastavení vlastností v daném adresáři. Jednotlivé volby se mohou kombinovat, přičemž vyhodnocování vlastností probíhá zleva doprava.
  - All .....Povoluje všechny vlastnosti (mimo vlastnosti MultiViews).
  - None.....Zakazuje vše.
  - ExecCGI..... Povoluje provádění CGI skriptů.
  - Includes..... Povoluje používat serverem vkládané vsuvky (SSI).
  - IncludesNOEXEC.....Totéž co předchozí, ale je zakázáno používat SSI direktivy #exec a #include, které se odkazují na CGI skripty.
  - FollowSymLinks.....Povoluje používat symbolické odkazy.
  - SymLinksIfOwnerMatch..... Totéž co předchozí volba, avšak vlastník odkazu i vlastník cílového souboru či adresáře musí být tentýž uživatel.
  - Indexes..... Nenachází-li se v daném adresáři výchozí dokument, který je definován v direktivě DirectoryIndex, je zobrazen obsah tohoto adresáře.
  - MultiViews.....Nastavení kódování na základě komunikace prohlížeče s WWW serverem.
- AllowOverride..... Specifikuje typy direktiv, které mohou být změněny direktivami souboru .htaccess. Volby jsou následující:
  - All ..... Lokální konfigurace překryje globální.
  - None.....Lokální konfigurace jsou ignorovány.
  - AuthConfig..... Lze nastavit individuální přístupová práva uživatelů
  - FileInfo..... Lze definovat charakteristiky souborů (kódování, jazyk, MIME typ).
- Alias.....Vystavení dokumentů z jiného adresáře než DocumentRoot.
- ScriptAlias.....Stejný význam jako předchozí, ale týká se umístění spustitelných souborů (CGI skriptů).
- UserDir.....Určuje jméno uživatelského adresáře, ve kterém může každý uživatel operačního systému mít uložené své vlastní stránky. Jedná se o podadresář, který je vytvořen v domovském adresáři uživatele, například public\_html.
- Redirect..... Přesměrování v případě, že dané dokumenty byly přesunuty na jiný server.

## Konfigurace typů souborů

- DefaultType.....implicitní MIME typ souboru, který server použije v případě, že nerozpozná typ souboru v požadavku. Výchozí hodnotou je obyčejný textový soubor, tedy MIME typ text/plain.
- AddType.....Definuje explicitně MIME typ souboru podle jeho přípony.
- AddLanguage.....Definuje příponu souboru použitou k rozlišení jazyka.
- AddEncoding.....Definuje příponu souboru použitou pro různé druhy kódování (šifrovací software, kompresní programy ap.), například: x-gzip, x-compress.
- LanguagePriorit.....Definuje pořadí jazyků, které jsou upřednostňovány.

## Konfigurace přístupových práv

- Deny.....Zákaz přístupu pro uvedené počítače. Může se zadat jméno počítače, IP adresa (i neúplná) nebo jméno domény (i neúplné).
- Allow.....Povolení pro uvedené počítače. Opět se může zadat jméno počítače, IP adresa (i neúplná) nebo jméno domény (i neúplné).
- Další parametry pro Allow a Deny:
  - all.....Povolení přístupu všem.
  - none.....Zákaz přístupu všem.
- Order.....Definuje pořadí platnosti direktiv Deny, Allow a Mutual-failure (povolení přístupu klientům, kteří jsou uvedeni v Allow a nejsou uvedeni v Deny).
- <Limit>.....Nastavení omezení pro konkrétní metodu protokolu HTTP:
  - GET.....Přenos dat, informace se přenáší jako součást URL za znakem ?, parametry se oddělují znakem &.
  - POST.....Přenos dat, pro předávání parametrů je použit nezávislý přenos, například z HTML formuláře.
  - PUT.....Přenos informací na server.
  - DELETE.....Zrušení informací na serveru.

## Nastavení přístupu konkrétním uživatelům

Přístup konkrétních uživatelů lze vymežit následující direktivou:

- **Require.....** Specifikuje, který uživatel má povolený přístup. Přístup je ověřen zadáním odpovídajícího jména uživatele a správného hesla. Jména uživatelů a jejich hesel, stejně tak jména skupin uživatelů, jsou uloženy ve zvláštních souborech (viz dále). Hodnoty direktivy Require jsou následující:
  - **User.....** Pouze uvedení uživatelé smějí využívat informace. Seznam uživatelů se odděluje mezerami.
  - **Group .....** Jen uvedené skupiny uživatelů smějí využívat informace. Seznam skupin se opět odděluje mezerami.
  - **Valid-user....** Přístup je povolen libovolnému uživateli, který je registrován v souboru hesel.
- **Autotype.....** Určuje typ autorizace. Výchozím typem je Basic.
- **AuthName.....** Text vložený za tuto direktivu se zobrazí uživateli současně s výzvou k zadání jména a hesla
- **AuthUserFile.....** Definuje soubor s hesly pro konkrétní uživatele. Jméno souboru se udává absolutní cestou.
- **AuthGroupFile .....** Definuje soubor skupin uživatelů. Jméno souboru se udává absolutní cestou.

Soubor se jmény a hesly uživatelů vytváří a spravuje program `htpasswd`. Informace v souboru hesel jsou uloženy ve tvaru `jméno:heslo`, heslo se generuje pomocí programu `htpasswd`. Definice uživatelů ve skupinách se zapisují do samostatného souboru ve tvaru `skupina:uživatelé`, kde uživatelé je seznam uživatelů ze souboru hesel oddělených mezerami. Všechny tyto soubory by měly být uloženy mimo obvyklý adresářový strom dokumentů.

Program `htpasswd`, který je uložen v adresáři `bin`, který je podadresářem adresáře s instalací Apache. Základní syntaxe je:

*`htpasswd [-cmdpsn] soubor _s_hesly jméno _uživatele`*



Parametry *htpasswd* příkazu jsou:

-c	Vytvoření souboru s hesly. Použijete pouze při prvním vytvoření souboru, jinak dojde přepsání existujícího souboru.
-m	Kryptování hesla pomocí MD5 (implicitní v prostředí Microsoft Windows ).
-d	Kryptování hesla pomocí CRYPT (implicitní v prostředí Unix a Linux).
-p	Heslo bude uloženo jako obyčejný text.
-s	Kryptování hesla pomocí SHA.
-n	Změny nebudou uloženy, pouze zobrazeny.

*Tab 1. Parametry souboru Htpasswd*

Ukázka vytvoření souboru `/etc/passwd.apache` a vložení uživatele posmura:

```
# htpasswd -c /etc/passwd.apache posmura
New password:
Re-type new password:
Adding password for user posmura
```

Příklad zabezpečení adresáře `C:/phpmyadmin/`, který je zpřístupněn z prostředí WWW jako

<http://localhost/phpadmin>

```
Alias phpadmin "C:/phpmyadmin/"
<Directory "C:/phpmyadmin/">
    Options MultiViews
    AllowOverride None
    AuthType Basic
    AuthName "Administrace databází"
    AuthUserFile "/etc/passwd.apache"
    Require User posmura
    Order allow,deny
    Allow from all
</Directory>
```

V tomto příkladu se smí připojit pouze uživatel posmura. Přístup tohoto uživatele bude ověřen zadáním hesla. Ostatní uživatelé budou odmítnuti.

- **Konfigurace virtuálních serverů**

WWW server Apache umožňuje konfiguraci více virtuálních WWW serverů. Virtuální servery můžete vytvářet jak na počítačích s více síťovými rozhraními, která mají vlastní IP adresy, tak na počítači s jediným síťovým rozhraním, které má přiděleno více IP adres. Jméno virtuálního serveru musí být definováno v DNS.

Příklad:

```
# Definice jména IP adresy virtuálního serveru
NameVirtualHost 212.11.122.11

# Definice virtuálního serveru
<VirtualHost 212.11.122.11>
ServerAdmin webmaster@posmura.taborsko.cz
DocumentRoot /home/web/posmura
ServerName posmura.taborsko.cz
ErrorLog /var/logs/ posmura.taborsko.cz.error_log
TransferLog /logs/ posmura.taborsko.cz.transfer_log
</Virtual>
```

Více informací naleznete na

<http://httpd.apache.org/docs-2.0/vhosts/>

### **2.2.3. Spuštění Apache v prostředí Microsoft Windows**

Existují dva způsoby spouštění WWW Serveru Apache:

#### **a) Start Apache jako služby**

V prostředí operačních systémů MS Windows NT a Windows 2000 je možné spouštět Apache jako službu operačního systému. Nastavení, zda se má služba Apache spouštět automaticky při startu operačního systému nebo ručně, popřípadě pozastavení a opětovné spuštění služby, provedete standardním způsobem, například ve Windows 2000:

Start\Nastavení\Ovládací panely\Nástroje pro správu\Služby

Další možností správy služby je pomocí nabídky

Start\Programy\Apache HTTP Server 2.0\Control Apache Server

kde je k dispozici volby *Restart*, *Start* a *Stop* služby Apache.

#### b) Start služby Apache z příkazového řádku

Pokud není potřeba mít WWW server Apache aktivní při každém spuštění počítače lze jej spouštět z příkazového řádku. Pro řízení služby je pak používán binární soubor *apache.exe*, který je uložen v podadresáři *bin* adresáře s instalací WWW serveru Apache, nebo můžete použít příkaz operačního systému *NET*. Parametry binárního souboru *apache.exe* získáme z nápovědy použitím *apache -h*.

## 2.3. PHP

PHP je freewarový skriptovací jazyk vhodný pro dynamické generování webových stránek. Velká výhoda PHP spočívá v jeho nezávislosti na používaných platformách. Výsledkem běhu PHP skriptu je obyčejný HTML dokument, který umí zobrazit každý prohlížeč. Odpadají tedy problémy s kompatibilitou, které přináší například Flash, PHP je nezávislé i na platformě serveru a může běžet pod libovolným webovským serverem pracujícím v operačním systému Windows nebo Unix.

### 2.3.1 PHP instalace

Instalační soubory jsou k dispozici k volnému stažení z adresy

<http://www.php.net/>

Máme na výběr archiv ZIP, či spustitelný instalační soubor. Soubor tedy rozbalíme, či nainstalujeme, např. do adresáře C:\php. Samotnou instalaci, ale nenastavíme vykonávání PHP z Apache. Máme dva způsoby spouštění PHP, jako interpret CGI, či jako zásuvný modul Apache. Pokud nechceme nastavovat konfigurační soubory sami, můžeme použít PHPTriad. Tento balík programů nám nainstaluje k Apache server PHP a MySQL a navíc sám nastaví konfigurační soubory. PHP se pak bude spouštět jako interpret CGI.

Správnou činnost ověříme vložením některé funkce PHP do HTML kódu, např.

```
<?php phpinfo();?>
```

### ➤ PHP jako interpret CGI

Prvním variantou spouštění PHP je varianta spouštění jako interpreta CGI skriptů. Interpretem, který bude provádět PHP skripty, je binární soubor *php* (v prostředí operačních systémů Unix) nebo *php.exe* (v prostředí Microsoft Windows). Z pohledu Apache se jedná o běžný CGI skript, to znamená externí program.

Prvním krokem je vytvoření virtuálního adresáře s CGI skripty:

```
ScriptAlias /php/ "c:/php/"
```

Druhým krokem je nastavení MIME typu pro soubory se skripty. Ty mají standardně příponu *.php*:

```
AddType application/x-httpd-php .php
```

Třetím a posledním krokem je přiřazení akce PHP skriptu:

```
Action application/x-httpd-php "/php/php.exe"
```

Samozřejmě cesty upravíme tak, aby odpovídaly adresáři nainstalovaného PHP.

### ➤ PHP jako modul Apache

Tato konfigurace je určena pro PHP, které běží jako modul Apache. Modul zavedeme v konfiguračním souboru Apache *httpd.conf* pomocí direktivy *LoadModule*. Binární instalace v prostředí Microsoft Windows je soubor s modulem PHP pro Apache umístěn v podadresáři *sapi* kořenového adresáře PHP.

```
LoadModule php4_module c:/php/sapi/php4apache.dll
```

V případě instalace PHP kompilací (viz výše) bude modul uložen v podadresáři *modules/* kořenového adresáře Apache:

```
LoadModule php4_module modules/libphp4.so
```

Nyní nastavíme MIME typ pro soubory se skripty. Ty mají standardně příponu *.php*.

```
AddType application/x-httpd-php .php
```

## 2.4. Matlab

Samotný Matlab není potřebné nikterak nastavovat, jen je nutné nainstalovat *MWS* a *Image Processing Toolbox*, či případné další funkce a toolboxy, které chceme využívat.

### 2.4.1. Nastavení MWS

Do adresáře *cgi-bin* nahrajeme spustitelný interpret *matweb.exe* a konfigurační soubor *matweb.conf*. Každá MWS aplikace musí mít v tomto souboru záznam.

Příklad záznamu v *matweb.conf*.

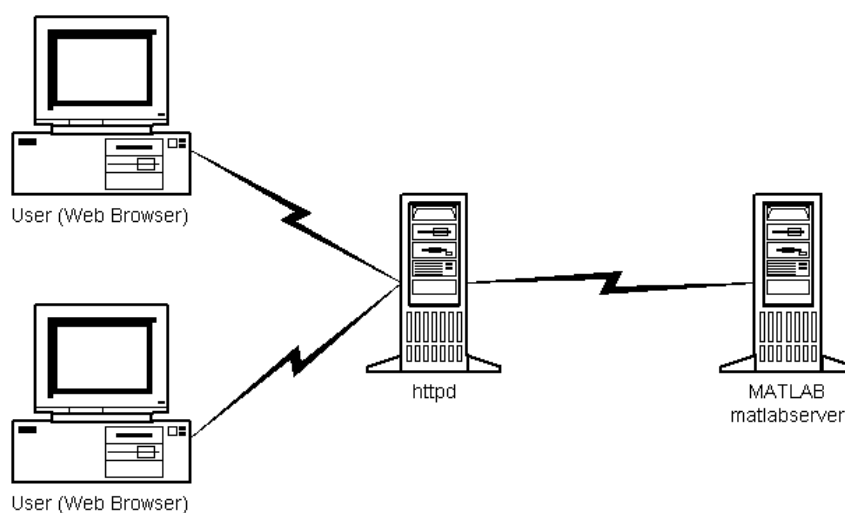
Položky v <i>matweb.conf</i>	Vysvětlení významu.
[diplomka]	Jméno souboru se zdrojovým kódem Matlabu bez přípony (soubor <i>diplomka.m</i> ).
mlserver=localhost	Jméno serveru.
mldir=c:\www\!diplomka	Adresář umístění aplikace.
mltimeout=1000	Max. čas výpočtu v ms.

Tab 2. Parametry nastavení *mathweb.conf*

Dalším konfiguračním souborem je *Matlabserver.cfg*. Význam parametrů uvádím v tabulce.

Parametr	Vysvětlení významu.
m 2	Počet vláken Matlabu.
p 8888	Port <i>matlabserver.exe</i> .

Tab 3. Parametry nastavení *matlabserver.cfg*



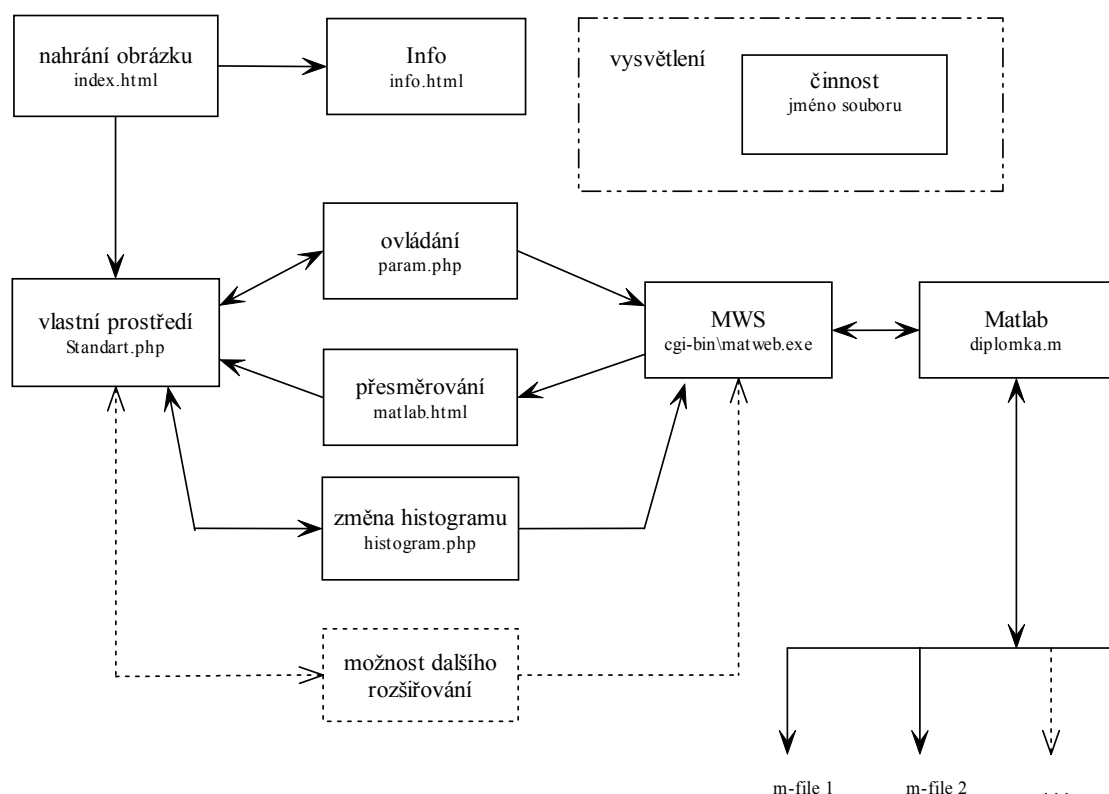
Obr 1. Způsob komunikace Matlab Web Serveru

### 3. Popis aplikace pro zpracování obrazu

#### 3.1. Popis způsobu komunikace

a) **Spuštění:** Po zadání URL adresy v internetovém prohlížeči je spuštěn soubor *index.php*. Tento soubor zjišťuje existenci pomocného souboru, který obsahuje data potřebná pro výpočty a informace o zpracovávaných souborech. Pomocný soubor je generován z IP adresy uživatele. Po zadání požadovaných údajů je soubor *index.php* vyhodnotí a pokud jsou splněny všechny náležitosti vytvoří nové okno prohlížeče, které se odkazuje na soubor *standart.php*.

b) **Práce v aplikaci:** Při výběru některé funkce se volá soubor *param.php*, který kontroluje, zda jsou všechny potřebné vstupy zadány. Při nesprávně zadaném vstupu je aplikace přesměrována zpět na *standart.php* s parametrem o chybě vstupu. Pokud jsou vstupy správně zadané voláme Matlab Web Server s parametrem prováděné akce. Popis komunikace Matlab Web Serveru je uveden níže. Po skončení výpočtu se spouští *matlab.html*, který zajišťuje návrat zpět na *standart.php*. Přímé přesměrování není možné z důvodu omezení MWS, který nespolupracuje s PHP.



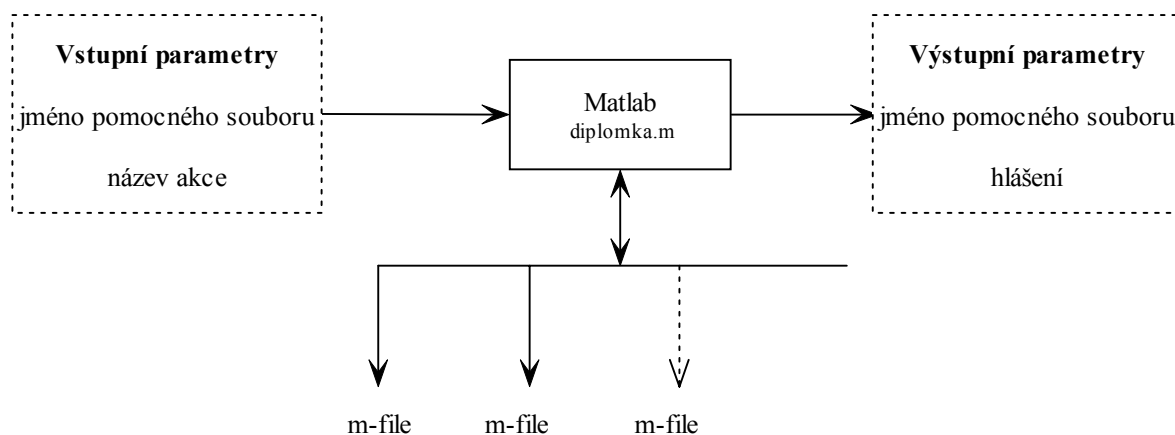
Obr 2. Blokové schéma směrování aplikace

### 3.2. Hlavní funkce matlab

Jednoduchým aplikacím využívající Matlab Web Server postačí jediný skript v Matlabu, který zajišťuje vstupy i výstupy. Pokud však aplikace využívá více funkcí je vhodné vytvořit řídicí skript. Ten pak zpracovává vstupní parametry, volá zvolené funkce a jejich výstupy vhodně upravuje a odesílá získané hodnoty jako výstupní parametry. Vytvoření řídicího skriptu rovněž vede ke zpřehlednění a zjednodušení vlastní struktury.

Hlavní skript *diplomka.m* má dva vstupní parametry, což je zřetelně vidět v následujícím obrázku. První parametr je jméno pomocného souboru, který obsahuje další proměnné a druhý pak obsahuje název akce, která se má provést.

Skript *diplomka.m* nejprve zjistí proměnné parametry z pomocného souboru. Poté se vygeneruje jméno dalšího obrazového souboru. Z důvodů použití cache (vyrovnávací paměti) nelze přepisovat původní soubor, ale je nutno generovat nové jméno souboru. Dále se v závislosti na vstupním parametru spustí požadovaná akce. Po vykonání se výsledek uloží pod vygenerovaným jménem a pomocný soubor se přepíše novými údaji. Výstupními parametry jsou *jméno pomocného souboru* a *zpráva*. Výstupní parametr *zprava* otevře v aplikaci okno s textem zprávy. Toho se využívá např. ve funkci rozrůstání oblasti, kdy parametr *zprava* obsahuje údaj o počtu bodů v oblasti.



Obr 3. Blokové schéma činnosti hlavní funkce v Matlabu

### 3.3. Ovládání aplikace

#### 3.3.1. Spouštění aplikace

Aplikace je dostupná na internetové adrese:

<http://147.230.129.170/!diplomka/>

##### ➤ První spuštění

Po zadání správné adresy je uživatel dotázán, zda si přeje pracovat s obrazovým souborem uloženým na serveru, nebo zda chce použít soubor vlastní. Aplikace je omezena na obrazové soubory BMP a JPG. Volba jiného souboru vyvolá chybové hlášení. Po úspěšném zadání je v novém okně spuštěna vlastní aplikace.

##### ➤ Druhé a další

Pokud při posledním spuštění uživatel práci neukončil funkcí v menu má možnost pokračovat v předchozí práci. Při zadání „*nepokračovat v předchozí práci*“ jsou soubory vymazány a následný postup je stejný jako při prvním spuštění. Volba „*pokračovat v předchozí práci*“ pak spustí aplikaci v novém okně se soubory z předešlé práce.

##### ➤ Ovládání a práce v aplikaci

Aplikace se ovládá pomocí menu umístěného vpravo nahoře, viz. níže kap. [3.3.2.]. Menu je aktivováno přejetím kurzoru myši. Jednotlivé funkce spustíme kliknutím levého tlačítka myši. Výsledné obrázky lze ukládat kliknutím pravého tlačítka na obrázku a volbě „*Uložit obrázek jako ...*“, resp. „*Save Image As ...*“.

Při volbě „*uprava (Java applet)*“ v menu „*Histogram*“ se spustí Java applet. Pokud se nezobrazí (šedé okno v místě appletu), pak internetový prohlížeč nepodporuje funkce Java appletu. To lze odstranit instalací vhodných zásuvných modulů do prohlížeče, či volbou jiného grafického prohlížeče. Změny histogramu se provádí:

- Kliknutím levého tlačítka myši pro změnu hodnoty jedné složky.
- Kliknutím pravého tlačítka myši pro změnu hodnot mezi označeným bodem (značen silnou tečkou) a bodem v místě kliknutí (počítá se rovnice přímky).
- Stisknutím levého či pravého tlačítka myši a pohybem kurzoru myši nad požadovanou oblastí měníme hodnoty jednotlivých složek.

Údaj vlevo dole je informace o aktuální složce (stará a hodnota po budoucím kliknutí). Vyrovnání spustí equalizaci a převod změnu převodní charakteristiky.



### ➤ Ukončení

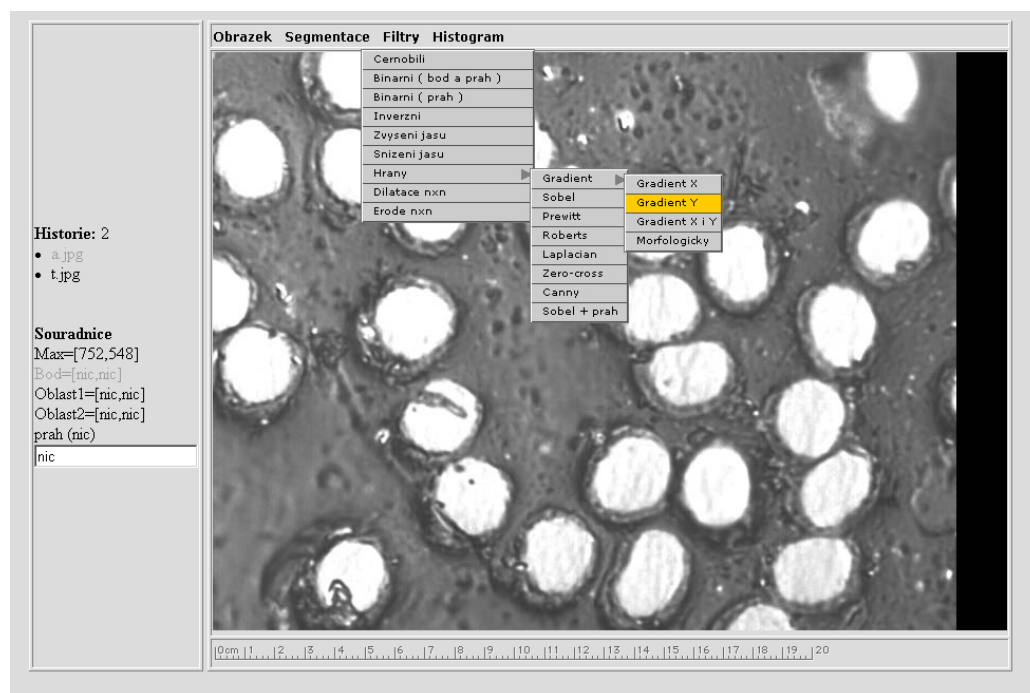
Ukončení aplikace můžeme provést zavřením okna aplikace. Při této volbě můžeme při dalším spuštění aplikace pokračovat v předchozí práci.

Další možností je použít funkce Ukončit v menu Obrázek. Touto volbou se smažou všechny zpracovávané soubory i pomocný soubor. Při dalším spuštění nemáme možnost pokračovat v předešlé práci.

### 3.3.2. Popis pracovní plochy

Popis pracovní plochy lze názorně vidět na obr 4. níže. Na levé straně máme historii práce s počtem zpracovávaných souborů, pokud si přejeme přejít zpět nebo dopředu k nějakému souboru, stačí na něj jednoduše kliknout na odkaz v levé části.

Pod historií se nachází oblast „Souradnice“. Položka „Max“ informuje o velikosti obrázku. Položky „Bod“, „Oblast1“ a „Oblast2“ značí, zda se po kliknutí na obrázek budou zaznamenávat souřadnice bodu, nebo oblasti a to podle toho, který údaj je aktivní (označen světleji). Jako poslední v levé části je kolonka pro zadávání hodnoty. Pravá horní část nabízí menu s jednotlivými funkcemi. Menu je aktivováno přejetím kurzoru myši a výběr funkce se provádí kliknutím levého tlačítka myši. Pravá dolní oblast je rezervována pro zpracovávaný obrázek.



Obr 4. Ukázka rozložení údajů v aplikaci

### 3.3.3. Položky v menu

Menu je realizováno pomocí Javaskriptu a DHTML. Z důvodů odlišnosti DHTML tagů jednotlivých prohlížečů je složité vytvořit takové menu, které by bezproblémově fungovalo pod všemi prohlížeči. Proto je nutné nejprve detekovat typ prohlížeče a jeho verzi a poté volat potřebné funkce. Podrobnější informace o tomto menu jsou v souboru *fw\_menu.js*. Jednotlivé položky menu jsou v souboru *standart.php*.

Ukázka jedné položky a její vysvětlení:

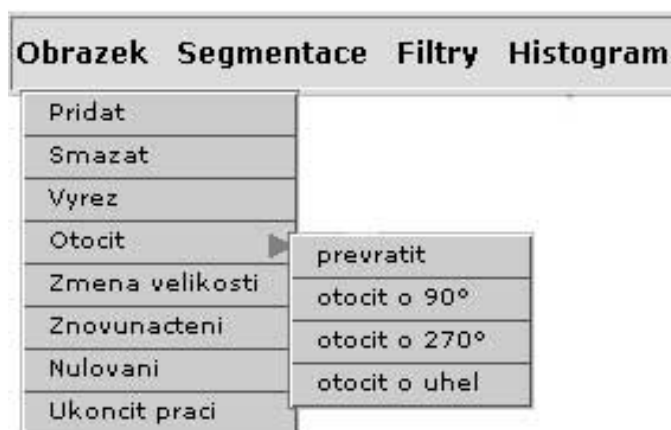
„fw\_menu\_0.addItem

("Pridat","location=\'\param.php?info\_file\_name=\'.\$info\_file\_name.'&akce=vyber\_soubor\'");

Popis: Do menu číslo 0 (číslování od 0, ta v našem případě odpovídá položce „Obrazek“) přidej položku s názvem „Pridat“, která se bude odkazovat na soubor uvedený za položkou „location=\\‘“. V našem případě se odkaz generuje pomocí kódu PHP. V odstavci 3.6 uvádím informace o rozšiřování aplikace.

V následujících bodech popíšu funkce jednotlivých položek menu a jejich požadované údaje.

#### a) Položka menu Obrazek

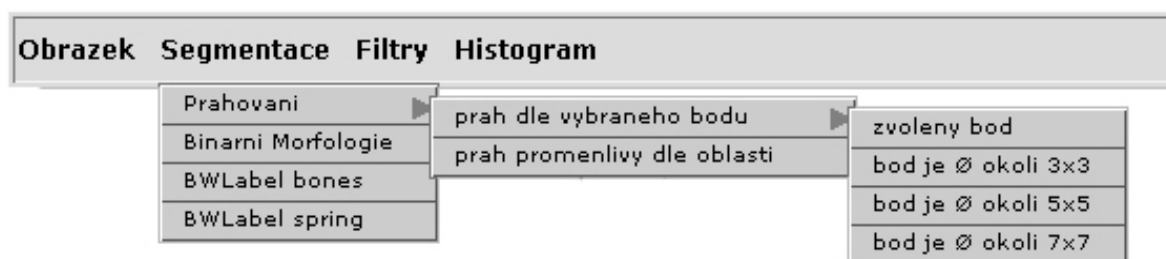


Obr 5. Ukázka menu Obrazek

<b>Priidat</b>	Přidá obrazový soubor do aplikace.
<b>Smazat</b>	Smaže aktivní obrazový soubor z aplikace.
<b>Vyrez</b>	Vyřízne oblast danou body Oblast1 a Oblast2. Požadované údaje: Oblast1 + Oblast2
<b>Otocit</b>	Podmenu.

<b>Prevratit</b>	Převrátí oblast (jako list papíru).
<b>Otocit o 90°</b>	Otočí obraz o 90°.
<b>Otocit o 270°</b>	Otočí obraz o 270°.
<b>Otocit o úhel</b>	Otočí o zadaný úhel po směru hodinových ručiček. Požadovaný údaj: Prah : otočení ve stupních
<b>Zmena velikosti</b>	Zvětší či zmenší na velikost zadanou v procentech. Požadovaný údaj: nová velikost obrazu v %
<b>Znovunactení</b>	Znovunačtení stránky.
<b>Nulovani</b>	Vynuluje všechny zadané údaje.
<b>Ukoncit praci</b>	Smaže obrazové soubory, konfigurační soubor a ukončí aplikaci.

b) Položka menu Segmentace



Obr 6. Ukázka menu Segmentace

- **Prahovani** Podmenu.
- **Prah dle vybraného bodu** Podmenu.
- **zvoleneny bod** Segmentace rozrůstáním oblasti, hraniční hodnota určena rozdílem jasu bodů, ten nesmí překročit prahovou hodnotu  
2 požadované údaje: Prah : prahová hodnota + Bod
- **Bod je Ø okoli 3x3** Segmentace rozrůstáním oblasti. Stejně jako předchozí, ale jas porovnávacího bodu určen průměrem 3x3 okolo zvoleného bodu.  
2 požadované údaje: Prahová hodnota + Bod
- **Bod je Ø okoli 5x5** Segmentace rozrůstáním oblasti, hodnota jasu zvoleného bodu je určena průměrem 5x5 okolo zvoleného bodu.  
2 požadované údaje: Prahová hodnota + Bod

- **Bod je Ø okoli 7x7** Segmentace rozrůstáním oblasti. Jas zvoleného bodu je dán průměrem okolí 5x5.  
2 požadované údaje: Prahová hodnota + Bod
- **Prah prom. dle oblast** Segmentace rozrůstáním oblasti, hraniční hodnota určena rozdílem jasu bodu zvoleného a testovaného. Hodnota prahu je spočítávána průměrem z bodů přidaných do oblasti a prvotním prahem.  
2 požadované údaje : Prahová hodnota + Bod
- **BWLabel bones** Segmentace bwlabel, výsledek v černobílých odstínech bones.
- **BWLabel spring** Segmentace bwlabel, výsledek v barvách spring.

c) Položka menu Filtry



Obr 7. Ukázka menu Filtry

- Cernobily** Převeďte obrázek do odstínů šedi.
- Binarni ( bod a prah )** Vytvoří binární obraz. Barva je určena ze zvoleného bodu +/- prah.  
2 požadované údaje: Prahová hodnota + Bod
- Binarni ( prah )** Vytvoří binární obraz. Komparační hodnota je hodnota prahu  
Požadovaný údaj: Prahová hodnota.
- Inverzni** Převeďte obrázek na inverzní (tj. maximální hodnota jasu – aktuální hodnota jasu).

<b>Zvyseni jasu</b>	Přičtení hodnoty jasu ke všem bodům. Požadovaný údaj: prahová hodnota
<b>Snizeni jasu</b>	Odečtení hodnoty jasu ke všem bodům. Požadovaný údaj: prahová hodnota
<b>Hrany</b>	Podmenu.
<b>Gradient</b>	Podmenu.
<b>Gradient X</b>	Detekce hran podle vzájemného poměru sousedních bodů ve směru osy X. Požadovaný údaj : prahová hodnota
<b>Gradient Y</b>	Detekce hran podle vzájemného poměru sousedních bodů ve směru osy Y. Požadovaný údaj : prahová hodnota
<b>Gradient X i Y</b>	Detekce hran podle vzájemného poměru sousedních bodů ve směru osy X i Y Požadovaný údaj : prahová hodnota
<b>Morfologicky</b>	Detekce hran pomocí morfologického gradientu Požadovaný údaj : prahová hodnota
<b>Sobel</b>	Detekce hran pomocí filtru Sobel.
<b>Prewitt</b>	Detekce hran pomocí filtru Prewitt.
<b>Roberts</b>	Detekce hran pomocí filtru Roberts.
<b>Laplacian</b>	Detekce hran pomocí Laplaceho filtru.
<b>Zero-cross</b>	Detekce hran pomocí filtru Zero-cross.
<b>Canny</b>	Detekce hran pomocí filtru Canny.
<b>Sobel+prah</b>	Detekce hran pomocí Sobel filtru s možností zadávat citlivost detekce. Požadovaný údaj : prahová hodnota
<b>Dilatace nxn</b>	Dilatace pomocí matice nxn. Požadovaný údaj: velikost matice n
<b>Erode nxn</b>	Erode pomocí matice nxn. Požadovaný údaj : velikost matice nxn

d) Položka menu Histogram



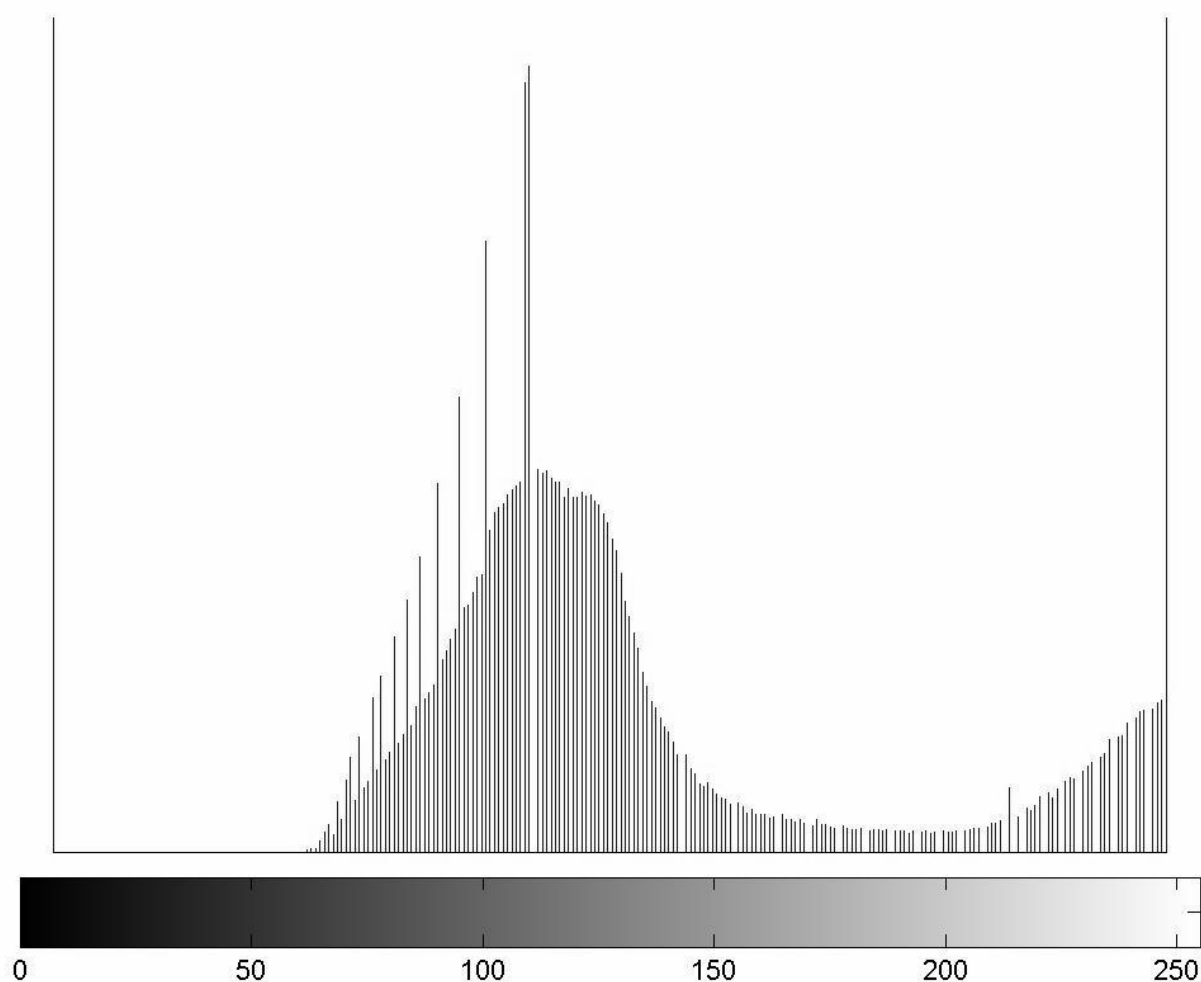
*Obr 8. Ukázka menu Histogram*

**Uprava ( java applet )**

Změna hodnot jasu jednotlivých složek pomocí Java appletu. Je vyžadována podpora Javy v prohlížeči. Ukázka níže na Obr. 14.

**Ukazat aktuální**

Ukáže histogram z aktuálního obrázku viz. následující obrázek.



*Obr 9. Ukázka histogramu*

### 3.4. Logování činnosti

Každá zvolená činnost se zaznamenává do souboru “log.log”. V tomto souboru je datum a čas spuštění, dále pak IP adresa uživatele a akce, kterou prováděl.

Ukázka souboru log.log

```
[10.5.2003 19:49:48] akce=logovani IP=147.230.154.98
[10.5.2003 19:50:36] IP=147.230.154.98 akce=gradient
[10.5.2003 19:50:41] IP=147.230.154.98 akce=prah
[10.5.2003 19:50:48] IP=147.230.154.98 akce=gradient
[10.5.2003 19:50:55] IP=147.230.154.98 akce=smazat
[10.5.2003 19:51:01] IP=147.230.154.98 akce=prah
[10.5.2003 19:51:06] IP=147.230.154.98 akce=gradient
[10.5.2003 19:51:11] IP=147.230.154.98 akce=smazat
[10.5.2003 19:51:15] IP=147.230.154.98 akce=prah
[10.5.2003 19:51:20] IP=147.230.154.98 akce=gradient
[10.5.2003 19:51:29] IP=147.230.154.98 akce=historie
[10.5.2003 19:51:33] IP=147.230.154.98 akce=historie
```

### 3.5. Omezení aplikace

- Nepřetržitý provoz MWS.
- Dostatek místa na disku serveru ( nastaveno 150 MB ).
- Maximální počet obrázků na uživatele ( nastaveno 15 ).
- Většina m-file pracuje v černobílém režimu.
- Nepodporuje českou diakritiku (nepodporuje Apache)

### 3.6. Možnost rozšiřování aplikace

Aplikace sice nebyla původně navrhována pro možnost dalšího rozšiřování, lze ji však rozšiřovat. Omezením může být vstup pouze jedné proměnné. Jeden vstup byl však pro naši aplikaci zatím dostačující a případné rozšíření nebude obtížné.

Postup rozšíření viz. Obr. 10 na následující straně. Zvolíme si hodnotu proměnné akce např. telefonuj. Změněny by měly být následující soubory:

#### ➤ Standart.php

Vyhledáme řádek : //sekce pro pridani dalsich polozek

Orientačně okolo řádku 120. Přečteme si příloženou nápovědu a podle návodu přidáme další menu, či jeho položky. V našem případě: do menu Filtry

```
fw_menu_2.addMenuItem(" Sem co se bude zobrazovat v menu  
", "location='param.php?info_file_name='.$info_file_name.'&akce=telefonuj'");
```

#### ➤ Param.php

Vyhledáme řádek //sekce pro pridani dalsich polozek.

Orientačně okolo řádku 500. Přečteme si příloženou nápovědu a podle návodu přidáme další menu, či jeho položky. V našem případě:

```
if ($akce == telefonuj)
{if (($oblastx1=="") || ($oblasty1=="") || ($oblastx2=="") || ($oblasty2==""))
{
header ("Location: ../!diplomka/standart.php?
info_file_name=".$info_file_name."&zprava=Sem se zadá co chceme sdělit
uživateli"); exit;
}
else
```



```

{
header ("Location: ../cgi-bin/matweb.exe?mlmfile=diplomka&
info_file_name=".$info_file_name."&akce=telefonuj");
exit;
}

```

➤ Diplomka.m

Vyhledáme řádek

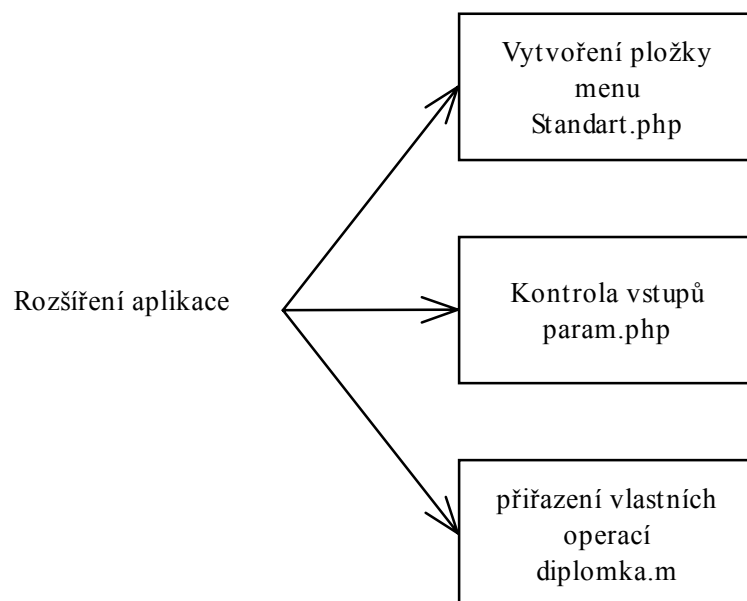
//sekce pro pridani dalsich polozek

Přidáme

```
case 'telefonuj'
```

```
if (size(im,3)~=1),im=rgb2gray(im);end;
```

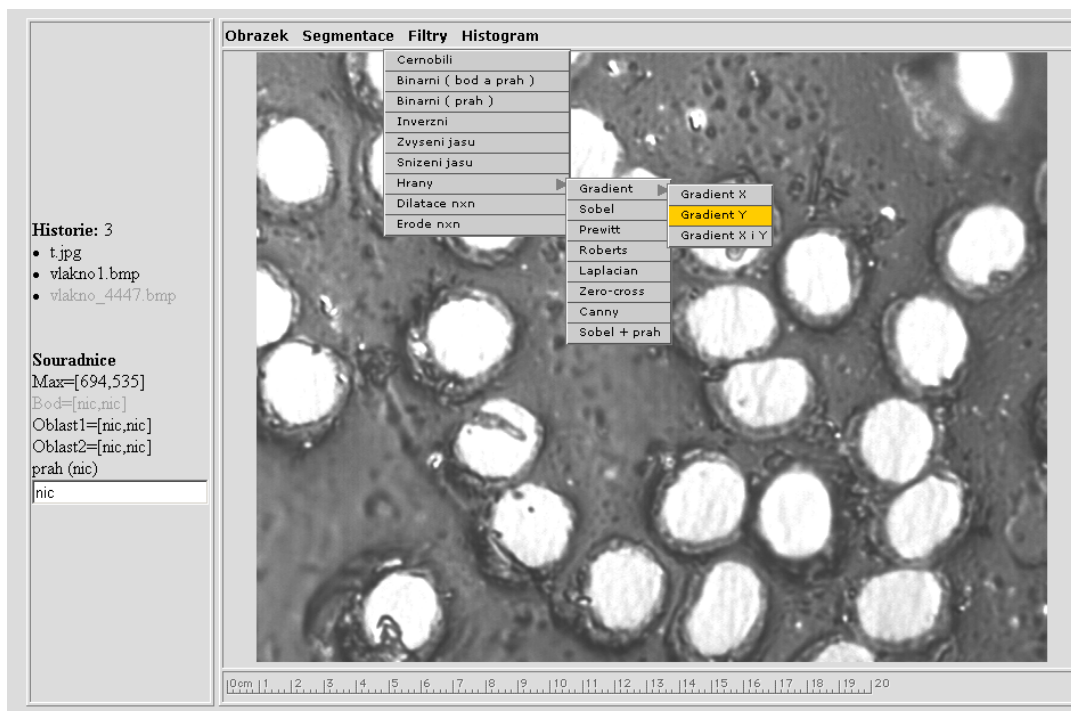
```
im = nekajafunkce;
```



*Obr 10. Blokové schéma rozšíření aplikace*

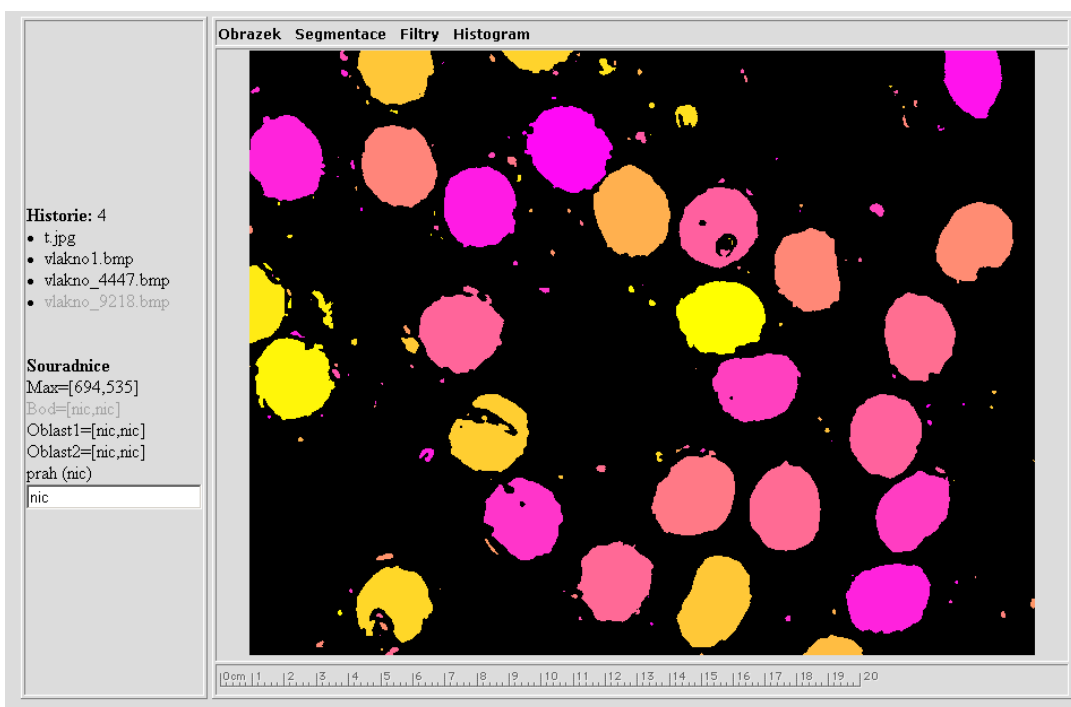
### 3.7. Ukázky aplikace

Ukázka prostředí, kde je vidět aktivní menu.



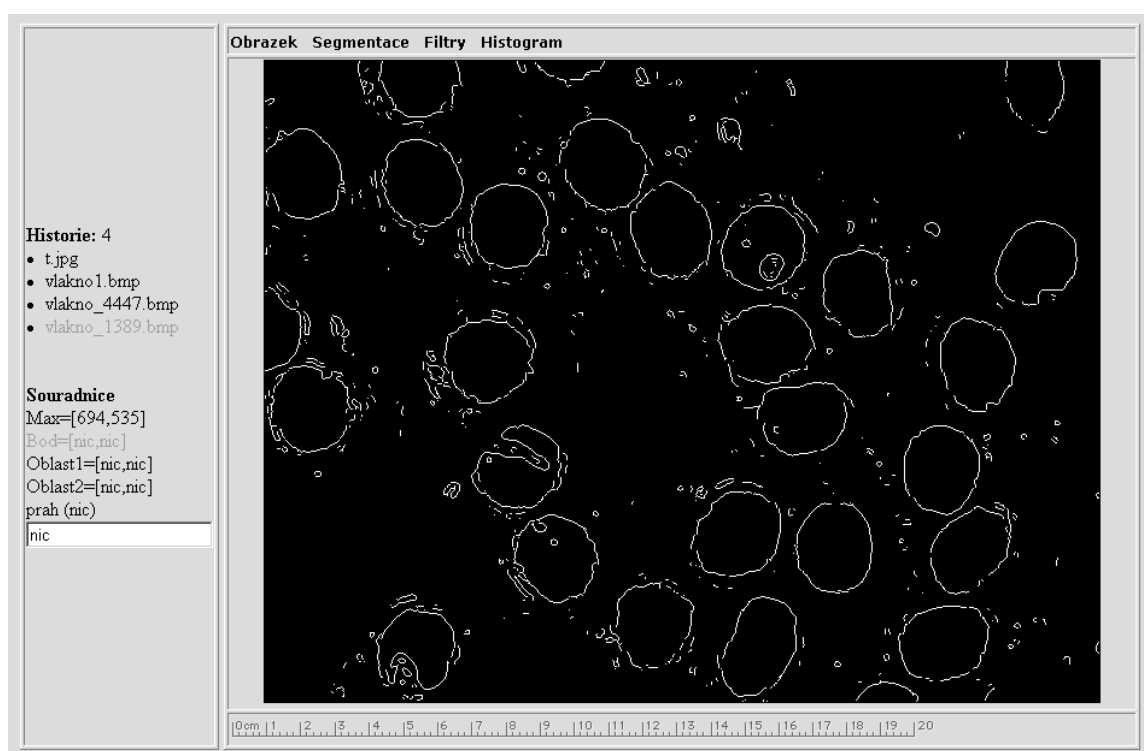
Obr 11. Ukázka aplikace 2.

Na následujícím obrázku vidíme výsledek funkce v menu segmentace BwLabel spring. V levé části (Historie) vidíme soubor *vlakno\_9218.bmp*, který byl přidán. Změnil se tedy i počet souborů v položce Historie.



Obr 12. Ukázka aplikace 2.

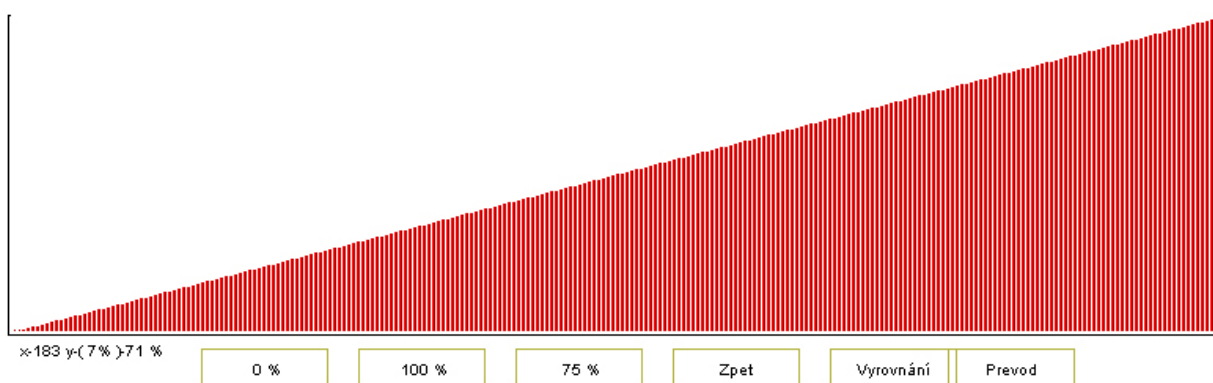
Ukázka po smazání souboru *vlakno\_9218.bmp* (údaje *Historie*) a volbě filtru typu Sobel.



Obr 13. Ukázka aplikace 3

Následující obrázek ukazuje Java applet. Svislé čáry představují:

- hodnoty úrovní histogramu, histogram obrázku bude vyrovnán na tyto hodnoty.
- převodní hodnoty histogramu, přičemž na ose X jsou vstupní hodnoty jasu a na ose Y hodnoty výstupní.



Obr 14. Ukázka aplikace 4

## 4. Řešení algoritmů

### 4.1. Předmluva

Jedním z důvodů výběru matlabu pro naši aplikaci byla skutečnost, že většina algoritmů pro zpracování obrazu je obsažena v Image Toolboxu, který jsem měl k dispozici. Rád bych dodal, že tyto algoritmy byly optimalizovány pro Matlab, a ty složitější jsou většinou spouštěny jako knihovny systému Windows, tudíž je jejich zpracování mnohem rychlejší.

### 4.2. Zpracování obrazu

#### Segmentace obrazu

Cílem je rozčlenit obraz do částí (objektů, segmentů) souvisejících s objekty reálného světa. Výsledkem je soubor vzájemně se nepřekrývajících oblastí. Rozlišujeme kompletní a částečnou segmentaci. Dokonalá segmentace vyžaduje znalost konkrétního řešeného problému.

Segmentace jsou založeny na globálních vlastnostech obrazu, nebo jeho částech (histogram). Dále možno segmentovat obraz určováním hranic mezi oblastmi (lokální informace jako jas, textura, gradient atd.), nebo přímým vytvářením hranic.

#### Některé druhy segmentací:

Segmentace prahováním je nejstarší a nejjednodušší.

Segmentace na základě detekce hran.

Segmentace vhodnou změnou histogramu.

Barevná segmentace.

Segmentace porovnáním se vzorem.

#### Předzpracování obrazu

Vhodným předzpracováním obrazu lze později obrázek lépe zpracovávat (segmentovat). Provádí se například transformací obrazu a aplikací vhodných filtrů (např. medián filtr pro odstranění šumu atd.).

### 4.3. Použité funkce Image Toolboxu

Použití těchto funkcí lze zjistit pomocí příkazu *help funkce*.

***Imread*** Slouží k načtení dat z obrazového souboru.

***Rgb2gray*** Funkce pro konverzi vícesložkových obrazových dat do jednosložkových.

***Im2bw*** Konverze obrázku do binárního obrazu.

***Bwlabel*** Segmentace porovnáváním sousedních bodů v matici.

***Label2rgb*** Vytvoří obrazové data z matice označených objektů.

<b>Edge</b>	Funkce na detekci hran podle mnoha kritérií.
<b>Imhist</b>	Zobrazení jednotlivých složek jasu.
<b>Histeq</b>	Vyrovnění jednotlivých složek jasu podle zadaných dat.
<b>Imwrite</b>	Uloží matici obsahující obrazová data do obrazového souboru.
<b>Imerode</b>	Erodovat obrazová data podle zadané matice.
<b>Imdilate</b>	Aplikuje dilataci na obrazová data podle zadané matice.
<b>Imrotate</b>	Otočí data matici podle zadaného úhlu.
<b>Imresize</b>	Změní velikost obrazové matice na zadanou hodnotu.

#### 4.4. Vytvořené funkce

Následující vytvořené funkce můžeme používat. Náповědu k těmto funkcím získáme v Matlabu pomocí *help funkce*.

- **barva1.m** Převod na binární obrázek pomocí zvoleného bodu s tolerancí

Použití:

```
Iout = barva1(Iin,y,x,tol);
Iin.....matice obrazových dat
Iout.....matice binárních obrazových dat
X.....hodnota souřadnice X
Y.....hodnota souřadnice Y
tol.....hodnota tolerance
```

- **barva2.m** Převod na binární obrázek, který získáme porovnáním bodů s hodnotou danou proměnou tol.

Použití:

```
Iout = barva2(Iin,tol);
Iin.....matice obrazových dat
Iout.....matice binárních obrazových dat
tol.....porovnávací hodnota
```

- **detekce51.m** Optimalizovaná segmentace rozšiřováním oblasti.

Použití:

```
[a, b] = detekce51(I, X, Y, Prah);
I.....matice obrazových dat
X.....hodnota souřadnice X
```

Y.....hodnota souřadnice Y

Prah....hodnota porovnávací úrovně

a.....výstupní matice rozpoznání objektu

b..... počet bodů v rozpoznané oblasti

➤ **detekce8.m**

Podobné jako Detekce51, ale vstupních souřadnic může být více a mohou být v matici.

Použití:

$[a, b] = \text{detekce8}(I, \text{body}, \text{Prah});$

I.....matice obrazových dat

Body....matice souřadnic bodu ve tvaru [y1 x1, y2 x2, ...]

Prah.....hodnota porovnávací úrovně

a.....výstupní matice rozpoznání objektu

b.....počet bodů v rozpoznané oblasti

➤ **GradientK**

Detekce hran pomocí gradientních funkcí.

Použití:

$I_{out} = \text{detekce8}(I, \text{prah}, \text{volba});$

Iin.....matice vstupních obrazových dat

Iout.....matice binárního obrazového souboru

prah.....hodnota porovnávací úrovně

volba.....další členění funkce

volba 1 - gradient ve směru osy X

volba 2 - gradient ve směru osy Y

volba 3 - gradient ve směru osy X i Y

volba 4 - morfologicky gradient

➤ **HistK.m**

Funkce na uložení dat histogramu do optimalizovaného obrazového souboru.

Použití:

$[I_{out}] = \text{HistK}(I_{in}, \text{jmeno});$

Iin.....matice vstupních obrazových dat

Iout.....matice optimalizovaných obrazových dat

jmeno.....jméno souboru, pod kterým se data uloží

➤ **HistK.m**

Funkce na změnu převodní charakteristiky histogramu.

Použití:

$[I_{out}] = \text{HistK}(I_{in}, \text{jmeno});$

*Iin*.....matice vstupních obrazových dat  
*Iout*.....matice optimalizovaných obrazových dat  
*jmeno*.....jméno souboru, pod kterým se data uloží  
 vytvoří inverzní obraz

➤ ***inverzním.m***

Použití:

$[ Iout ] = \text{inverzním.m} (Iin );$   
*Iin*.....matice vstupních obrazových dat (podpora vícesložkových obrazů)  
*Iout*.....matice optimalizovaných obrazových dat

➤ ***JasK***

Použití:

Změna jasu posunutím histogramu ve směru osy y.

$Iout = \text{jasK}(Iin, \text{prah}, \text{volba});$   
*Iin*.....matice vstupních obrazových dat  
*Iout*.....matice optimalizovaných obrazových dat  
*prah*.....hodnota posunutí  
*volba*.....další členění funkce  
*volba*.....1....zvýšení jasu  
*volba*.....2....snížení jasu

➤ ***Ksobel.m***

Použití:

Filtr Sobel s možností změny citlivosti detekce hran.

$Iout = \text{Ksobel}(Iin, \text{prah});$   
*Iin*.....matice vstupních obrazových dat  
*Iout*.....matice optimalizovaných obrazových dat  
*prah*.....citlivost, čím menší, tím větší citlivost

➤ ***otoc.m***

Použití:

Otočení matice o 180, nebo 270 stupňů.  
 $y = \text{otoc}(im, \text{kolik});$   
*Im*.....matice vstupních obrazových dat  
*y*.....matice otočených dat  
*kolik*.....volba otočení. Může být 180, nebo 270 stupňů.

- ***sdo.m*** Skript na segmentaci dělením oblasti.  
 Použití:  
     sdm  
     Důležité je, aby obrázek byl dělitelný  $2^n$ . To nastavujeme pomocí funkce *resize()* na začátku skriptu.
- ***sbm.m*** Segmentace pomocí funkcí Morfologie.  
 Použití:  
     *[Iout]=sbm(Iin, b)*  
     Iin.....matice vstupních obrazových dat  
     Iout.....matice výstupních obrazových dat  
     b.....segmentovací velikost (např. 20)
- ***sedy2.m*** Převod barevného na černobílý obrázek.  
 Použití:  
     *[Iout]= sedy2 (Iin)*  
     Iin.....matice vstupních obrazových dat  
     Iout.....matice výstupních obrazových dat
- ***sphp.m*** Segmentace složek histogramu prahováním.  
 Použití:  
     *Iout = sphp(Iim,prah,barva);*  
     Iin.....matice vstupních obrazových dat  
     Iout.....matice výstupních obrazových dat  
     prah.....hodnota tolerance  
     barva.....0 všechny největší ( prvních 10)  
     barva.....1,2,... vyber 1. nejvíce zastoupenou, 2. nejzastoupenější, ...

#### 4.5. Využívání DLL knihoven

Matlab funkce *mcc*, či *mex* nám dovoluje vytvářet knihovny Windows, tzv. DLL soubory. Tyto knihovny jsou rychlejší, než klasický *m-file*. Bohužel tyto nepodporují rekurzivní volání funkcí, které bych využil např. při metodě klasického prahování. Zkonvertováním vytvořených funkcí do těchto knihoven se výpočty urychlily.



## 4.6. Problémy při tvorbě matlab funkcí

Během tvorby se objevovaly nejrůznější problémy od jednoduchých po složitější, které bylo nutno vyřešit. Odstraňování těchto problémů bylo ještě složitější z důvodů práce po internetu. Internetový prohlížeč, podle svého nastavení, používá vyrovnávací paměť (cache) pro stáhnuté soubory a změny se projeví až po znovunačtení do této paměti. Hlavní problémy s vyrovnávací pamětí se podařilo odstranit generováním jedinečných názvů jmen souboru. Dále je vhodné vypnout cachování prohlížeče. Prohlížeč potom nebude nahrávat již starší verze z cache.

Jedním z problémů na které bych rád poukázal, je přetypování proměnných a kontrola správnosti typů. Vstupní proměnné jsou typu string. My většinou potřebuje typ double. Bylo nutno vyřešit i prázdné vstupy atd.

Dalším problémem bylo například zobrazení funkce histogramu. Tato funkce je v matlabu již implementována, ale při uložení těchto dat do obrazového souboru byla výsledná velikost souboru v MB, což je neúnosné. Při komprimaci souboru do černobílého obrázku ve formátu JPG byly odstraněny krajní hodnoty histogramu (z důvodů ohraničení dat). To bylo vyřešeno přeprogramováním vnitřní funkce.

Větší problém nastal při pokusu zautomatizovat segmentaci tak, že z obrázku nejprve segmentujeme nejvíce zastoupenou složku, pak druhou nejvíce zastoupenou atd. Dále máme možnost si vybrat jakou složku z nejvíce zastoupených chci segmentovat. Tento algoritmus sice funguje, ale pro segmentaci více složek je rychlost nedostačující, proto jsem jej nezačlenil do aplikace.

Pokus o algoritmus watershed dopadl podobně. Hranice oblasti jsem spočetl pomocí funkce morfologického gradientu. Ten je definován takto:

$$g(f) = (f \ominus B) - (f \circ B)$$

kde  $(f \ominus B)$  značí dilataci jednotlivých bodů a  $(f \circ B)$  je eroze jednotlivých elementů. Na výsledek se pak aplikuje algoritmus rozrůstání oblasti a „semínka“ jsou vloženy do oblastí s nejnižší hodnotou jasu. Z aplikace lze postup provádět postupně viz. Kapitola 5.

### **Pokusy s rozrůstáním oblasti:**

Metoda rozrůstání oblasti klasickým způsobem, kdy jsou body označené k porovnávání přidávány do fronty. Kód programu je v souboru *detekce7.m*. Algoritmus funguje, ale jeho rychlost se počítá na desítky sekund a navíc má velké nároky na paměť. Tento algoritmus jsem se pokoušel naprogramovat v jazyku C pomocí rekurzivního volání funkce, které, jak jsem posléze zjistil, není v DLL knihovnách podporováno. Další problém byl ve velikosti fronty

(uložené body určené ke kontrole). Ta se teoreticky mohla pohybovat při obrázku 1000x1000 bodů až 1000 x 1000 x velikost jedné položky fronty (např. 16 bitů) bitů, čili hodnota v MB. Problémy jsem nakonec vyřešil funkcí *detekce51.m*. Je složitá, protože neprochází celou maticí, ale jen ty části, které vyžadují kontrolu.

#### 4.7. Postupy

Odstranění šumu lze efektivně provést použitím vhodného filtru. V aplikaci je používán filtr Medián.

**Segmentace prahováním** můžeme použít v případech odlišnosti barev segmentů od okolí. Tento způsob je jednoduchý, velice rychlý, avšak výsledná segmentace objektů obsahující více barevných odstínů není dostatečná.

**Segmentace rozrůstání oblasti** je algoritmus, kdy je obraz segmentován z míst, kde vložíme tzv. semínka. Okolní body těchto semínek jsou kontrolovány, zda splňují podmínku. Pokud ji splňují, přidají se do segmentované oblasti a kontrolují se okolní body tak dlouho, dokud neexistuje další okolní bod splňující podmínku. V našem případě je podmínka ve tvaru, kdy požadujeme, aby rozdíl jasových hodnot byl menší, než stanovená hodnota. Výsledkem této funkce je matice rozpoznání objektu a počet bodů v oblasti viz Odstavec 4.4., či 5.2.

**Segmentace dělením oblasti:** Tento algoritmus vyžaduje rozměr matice dělitelný  $2^n$ . Jde o opačný postup, než při rozrůstání oblasti. Obrazová data jsou dělena na menší části tak dlouho, dokud není splněna podmínka jednotnosti jasových hodnot.

**Segmentace pomocí funkcí morfologických operací:** Na obrázek je aplikována eroze ve směru osy X. Na výsledek použijeme dilataci se stejnými parametry jako eroze. Analogicky pokračujeme ve směru osy Y. Výsledný obrázek je dán průměrem výsledků z obou os. Použití tohoto filtru je demonstrováno v tab. 4.

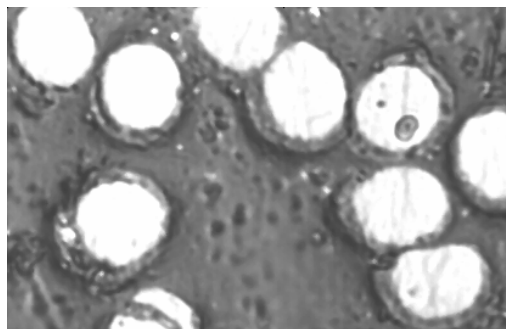
**Vhodnými transformacemi histogramu** lze obrazová data upravovat tak, aby případné další metody zpracování obrazu dosahovaly lepších hodnot. Můžeme např. měnit kontrast, celkový jas i provádět segmentaci prahováním atd. V aplikaci je využit java applet pro práci s histogramem. Máme možnost vyrovnávat jasové úrovně, nebo měnit převodní charakteristiky. Používání změn převodních charakteristik je vhodné z důvodů nenáročnosti výpočtů, přičemž jejich aplikací lze docílit požadovaného výsledku. Demonstrace postupů změn histogramu jsou v kapitole 5.

## 5. Ukázky

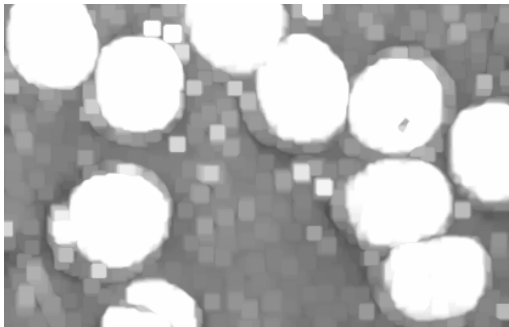
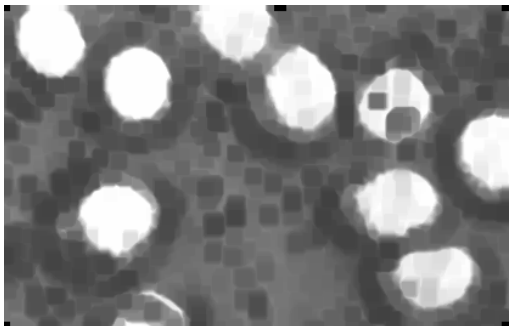
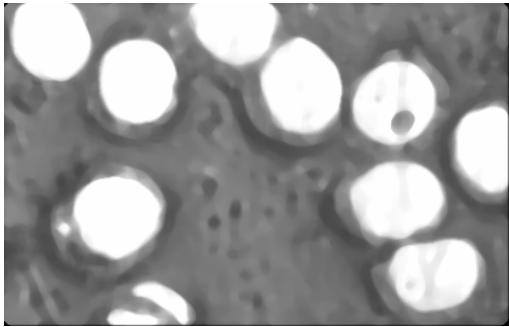
Tato kapitola ukazuje výsledky některých vybraných funkcí.


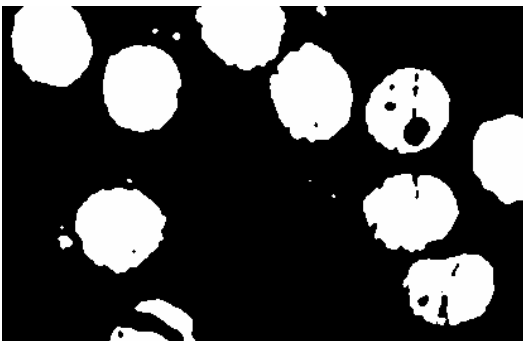
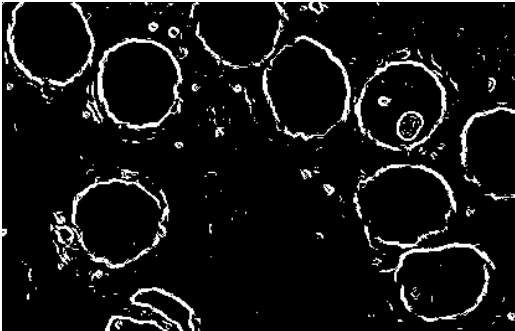
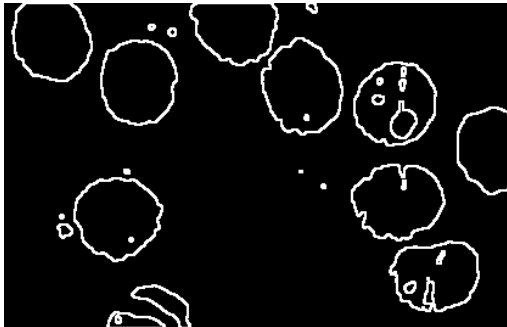
### 5.1. Ukázka 1

Demonstruje aplikaci různých typů filtrů na ukázkovém obrazovém souboru.



*Obr 15. Vstupní obrázek*

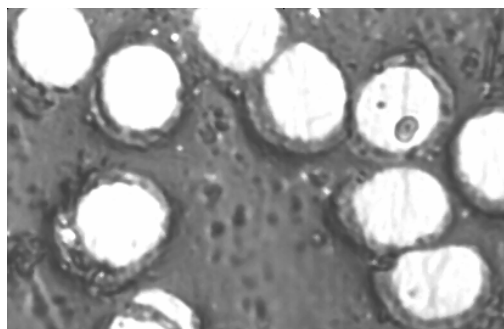
Filtr Erode nxn Prah = 10	
Filtr Dilate nxn Prah = 10	
Filtr Media nxn Prah = 10	

<p>Detekce hran Sobelův filtr</p>	
<p>Binární obrázek Binární (bod a prah) Prah = 20 Bod uvnitř bílé plochy</p>	
<p>Detekce hran Gradient X i Y Prah = 20</p>	
<p>Detekce hran Filtr binární obrázek Prah = 200 Filtr Hrany gradient Morfologicky</p>	


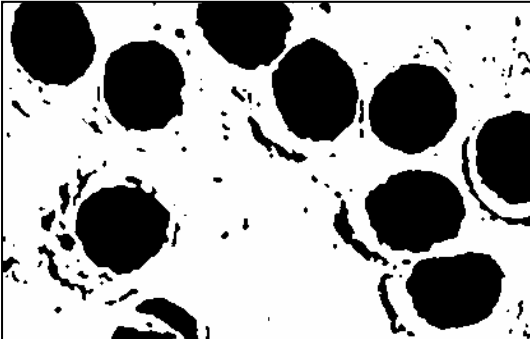

Tab. 4. Ukázka aplikace filtrů

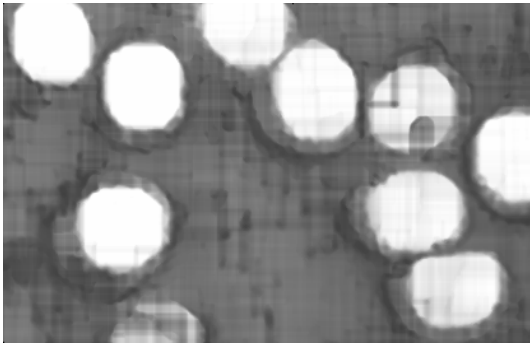
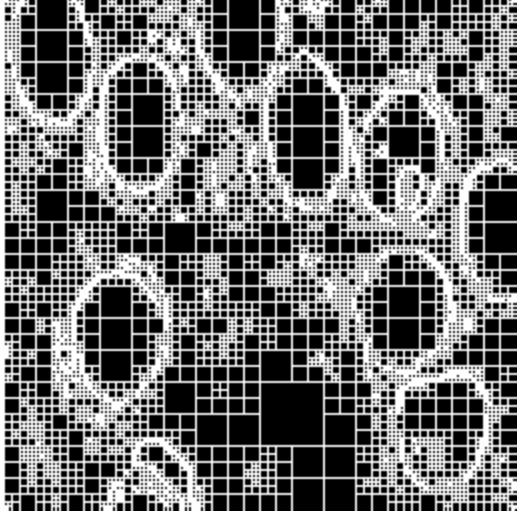
## 5.2. Ukázka 2

Demonstruje aplikaci různých způsobů segmentace obrazu.



Obr 15. Vstupní obrázek

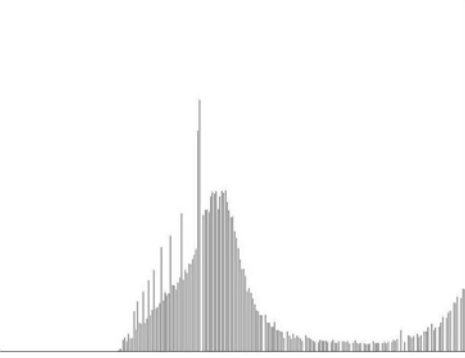
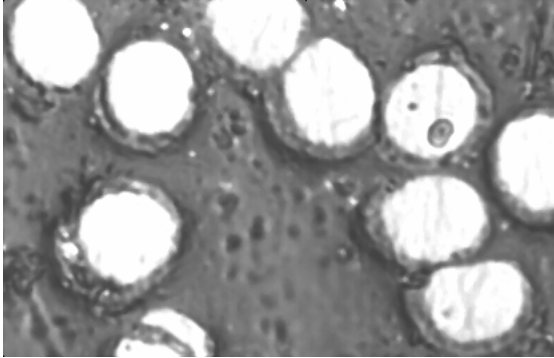
<p>Segmentace rozrůstáním oblasti  Bod = uvnitř bílého objektu  Prah = 20</p> <p>Počet bodů oblasti = 3 410</p>	
<p>Segmentace rozrůstáním oblasti  Bod = pozadí  Prah = 20</p> <p>Počet bodů oblasti = 73 023</p>	
<p>Postup popsáný jako „watershed“  binární obraz  Práh = 20  Morfologický gradient  Prah = 100  Segmentace rozrůstáním oblasti  Prah = 20  Bod = pozadí  Počet bodů oblasti = 77 418</p>	

Segmentace funkcemi morfologie	
Ukázka segmentace dělením oblasti Je patrná změna velikosti obrazu.	

*Tab. 5. Ukázky segmentací*

## 5.2. Ukázka 3

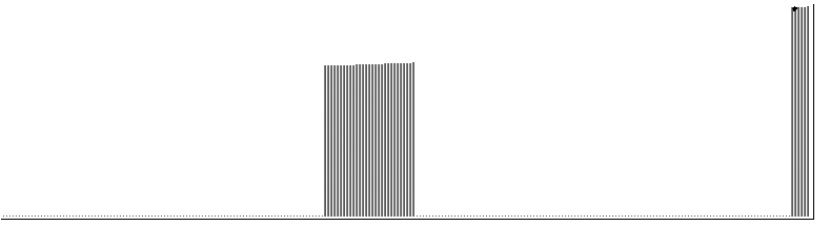
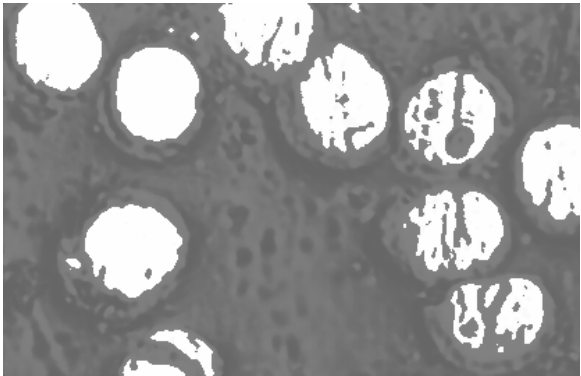
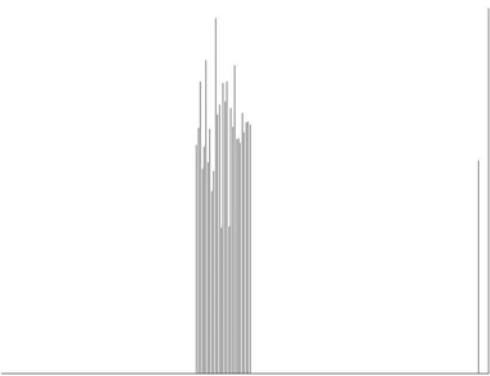
Následuje série příkladů, které ukazují vyrovnávání histogramu Příklad 1. – Příklad 4.

	
Původní histogram	Původní obrázek

*Tab. 6 Původní obrázek*

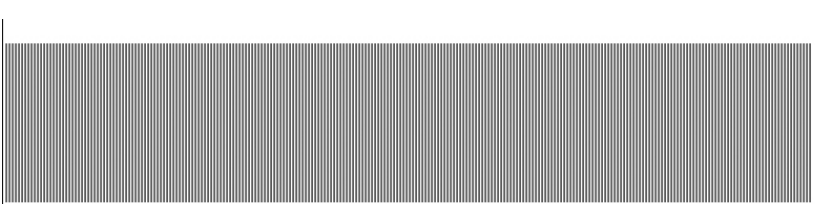
**Příklad 1.**

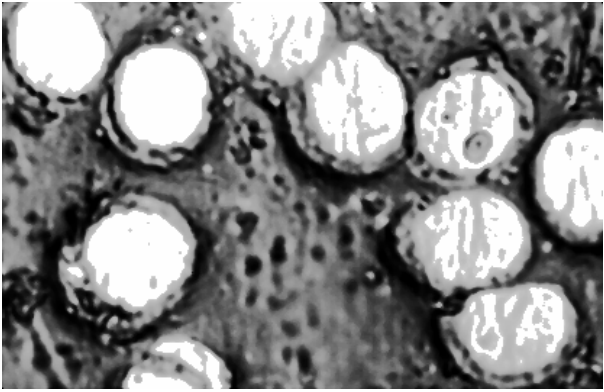
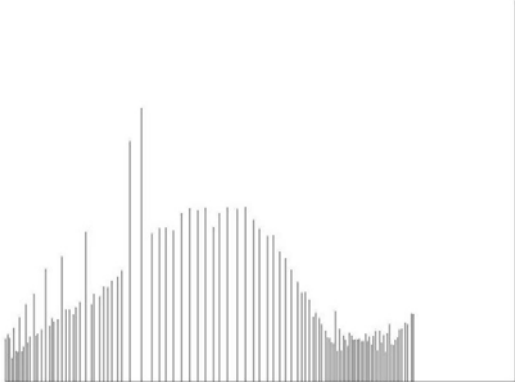
Pokus o segmentaci do dvou barev pomocí vyrovnání histogramu podle následujícího obrázku na původní obrázek.

Aplikované vyrovnání histogramu	
Výstupní obrázek	
Výstupní histogram	

*Tab 7. Příklad 1.*

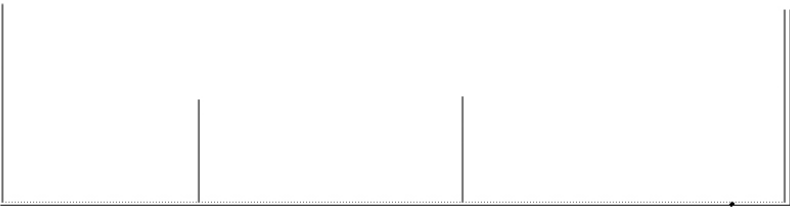
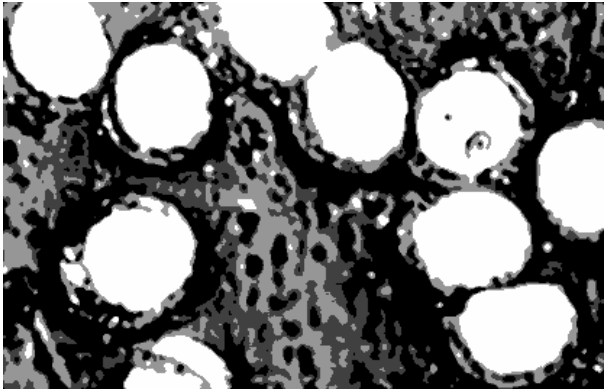
**Příklad 2.** Aplikace vyrovnání histogramu podle následujícího obrázku na původní obrázek.

Aplikované vyrovnání histogramu	
---------------------------------	--

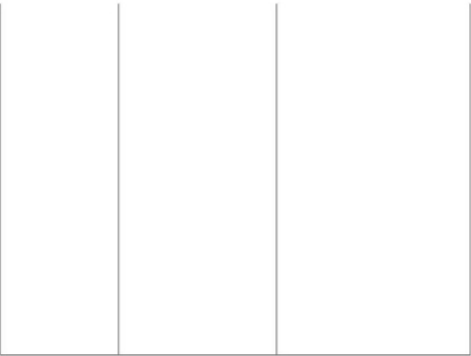
Výstupní obrázek	
Výstupní histogram	

Tab 8. Příklad 2.

**Příklad 3.** Pokus o segmentaci do čtyř barev pomocí vyrovnání histogramu podle následujícího obrázku na původní obrázek.

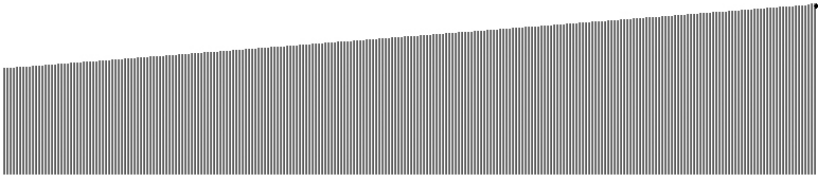
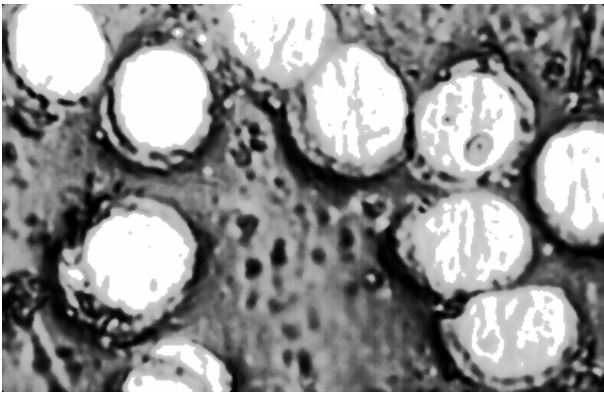
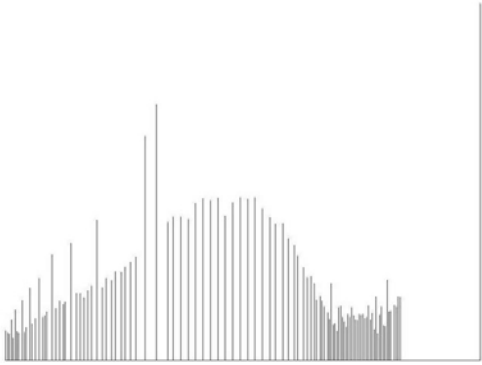
Aplikované vyrovnání histogramu	
Výstupní obrázek	



Výstupní histogram	
--------------------	--

Tab 9. Příklad 3.

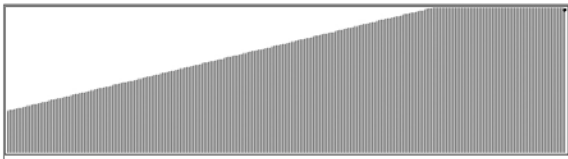
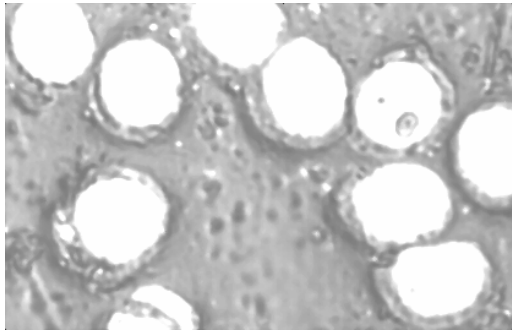
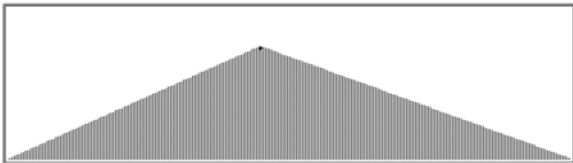
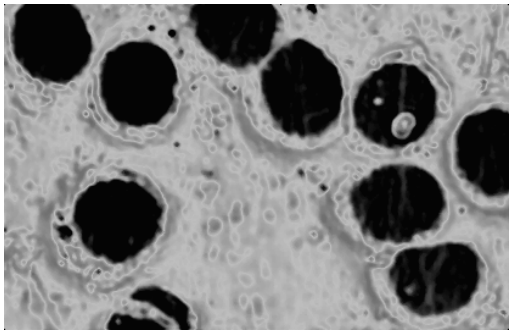
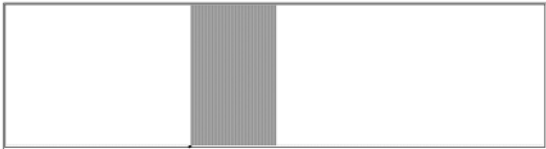
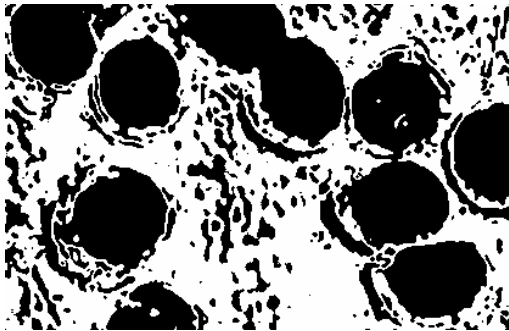
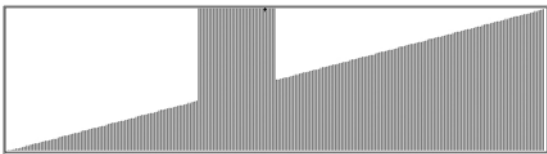
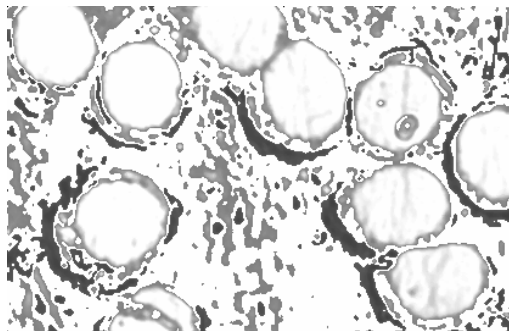
**Příklad 4.** Pokus o zvětšení kontrastu zvýšením úrovně jasu histogramu

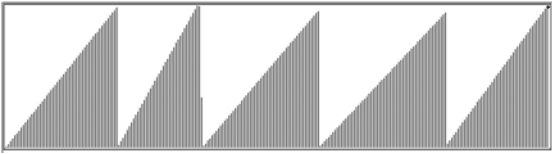
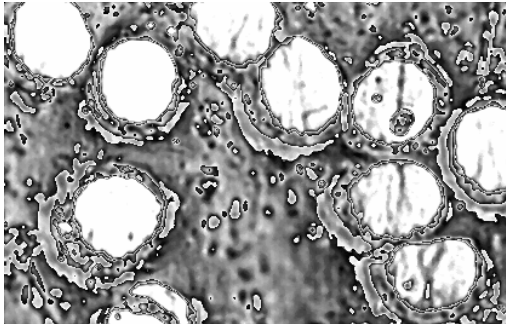
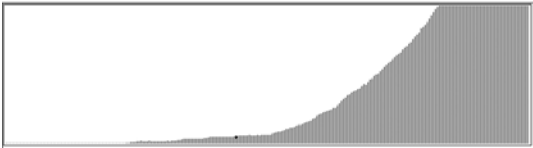
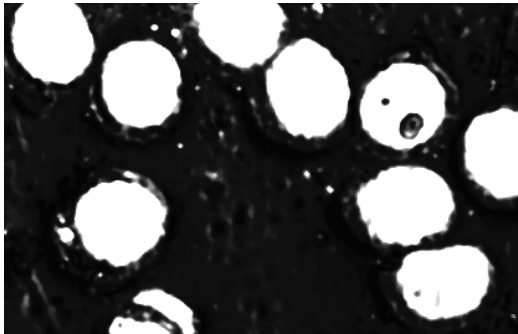
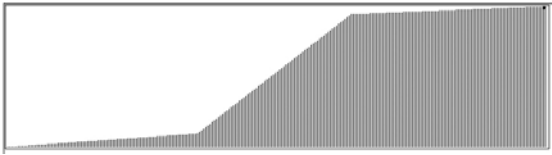
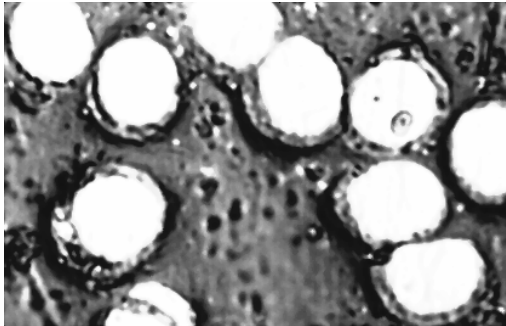
Aplikované vyrovnání histogramu	
Výstupní obrázek	
Výstupní histogram	

Tab 10. Příklad 4.

### Příklad 5.

V následující tabulce jsou ukázky na změny převodních charakteristik histogramu. Jako vstupní obrázek byl použit obrázek z předchozích příkladů viz. obr. 17.

<p>Úprava jasu (zvýšení)</p> 	
<p>Částečný negativ</p> 	
<p>Víceúrovňové prahování</p> 	
<p>Částečný originál a prahování</p> 	

<p>Přetečení dyn. rozsahu</p> 	
<p>Nelineární průběh</p> 	
<p>Menší změna</p> 	

Tab. 11. Příklad 5.

### Shrnutí k ukázkám:

Výsledky úprav obrazu filtracemi nepřinesly nic nového. Překvapením pro mě bylo použití filtru morfologického gradientu. Zde byly hranice objektů silně zřetelné. Toho využívá metoda segmentace nazvaná „watershed“.

Metoda vyrovnání histogramu nemá dobré výsledky, ať již samotné segmentace nebo pokusy transformovat obraz do podoby vhodnější k segmentaci. Jako nejzdařilejší pokus se mi jeví příklad 3., kdy jsem se pokoušel segmentovat obraz do čtyř základních barev.

Změny převodních charakteristik jsou časově nenáročné a mají velké možnosti použití viz. tab. 11.

## 6. Závěr

Programování náročných funkcí vyžaduje nejen znalost základních funkcí a klasických postupů programování, ale i detailní znalost časové náročnosti jednotlivých funkcí a možností využití optimalizovaných funkcí v Matlabu.

Zrychlení přineslo přepsání funkcí do knihoven Windows (DLL soubory) pomocí funkce *mcc*, resp. *mex*. Tyto knihovny jsou pak volány přímo jako samostatná aplikace, Matlab tedy není zatěžován interpretováním příkazů Matlab funkce.

I přes zrychlení těchto funkcí však není rychlost dostatečná na to, aby mohly být vytvořené funkce efektivně aplikovány v aplikaci.

Metody změny převodních charakteristik histogramu se ukázali jako vhodné řešení z důvodu nenáročnosti výpočtů.

Popsané algoritmy a postupy pro zpracování obrazu nevysvětlují její celou problematiku, spíše naznačují možné řešení. Dokonalé řešení vyžaduje znalost konkrétního řešeného problému, proto je obtížné vymyslet univerzální algoritmus.

Výsledná aplikace je rozšiřitelná nejenom o další Matlab funkce způsobem popsáním v kapitole [3.6], ale je možné přidávat i vlastní programy napsané např. v Javě.

## Literatura

- [1] Sonka M., Hlavac V., Boyle R., Image processin, Analysis and Machine Vision, International Thomson Publishing, Inc., Second edition, 1999.
- [2] KletteT., Zaperoni P., Handbook of Image Processing Operators, J. Wiley & Sons, 1996
- [3] Beucher S., Application of Mathematical Morphology in Material Science, 1995, pp. 41.-46.
- [4] Matlab – Image Processing Toolbox, v.3, MathWorks, Inc., (příručka uživatele).
- [5] Nouza J., Metody rozpoznávání, přednášky 2001
- [6] Šilhán D., Zpracování signálů a obrazů (navštíveno 18.5.2003).  
URL: <http://ds.webzdarma.cz/set.php?g=fel-zso>
- [7] Košek J., On-line příručky a návody (aktualizováno 25.4.2003)  
URL: <http://www.kosek.cz/>
- [8] Pošmura V., Apache, PHP a MySQL (navštíveno 18.5.2003).  
URL: [http://posmura.taborsko.cz/php/apache\\_php\\_mysql.htm](http://posmura.taborsko.cz/php/apache_php_mysql.htm)
- [9] Matlab Web Server, FEL CVUT (navštíveno 18.5.2003).  
URL: <http://mws.felk.cvut.cz/>
- [10] Hozman J., FEL CVUT (navštíveno 18.5.2003).  
URL: [http://radio.feld.cvut.cz/courses/ZSL1/Zprac\\_obr\\_prisp\\_sb\\_M2002.pdf](http://radio.feld.cvut.cz/courses/ZSL1/Zprac_obr_prisp_sb_M2002.pdf)

## Příloha A

### Ukázky zdrojových kódů

#### ➤ Ukázka Matlab funkce

```
function out = sbm(im,prah);

%segmentace binarni morfologii
%pouziti
%[Iout]=sbm(Iin, b)
% Iout   vystupni image
% Iin    Vstupni image
% b      Segmentovaci velikost (napr. 20)

n=prah;
[ym,xm,rgb]=size(im);
if (rgb>1),im=rgb2gray(im);end;
b=ones(1,n);
im1=imerode(im,b);
im1=imdilate(im1,b);
b=ones(n,1);
im2=imerode(im,b);
im2=imdilate(im2,b);
im1=double(im1);
im2=double(im2);
for y=1:ym,
    for x=1:xm,
        im3(y,x)=round((im1(y,x)+im2(y,x))/2);
    end
end
out=uint8(im3);
```

➤ Ukázka z hlavní funkce diplomka.m

...

switch (akce)

case 'hrana1'

if (size(im,3)~=1),im=rgb2gray(im);end

[im,pom]=detekce51(im,y,x,prah);

zprava=strcat('pocet bodu v oblasti = ', num2str(pom));

case 'prahovani' %

if (size(im,3)~=1),im=rgb2gray(im);end;

im = detekce7(im,[y x;10 13],prah);

...

➤ Ukázka z řídicího skriptu param.php

...

if (\$akce == resize)

{

if (\$prah == "")

{

header ("Location:../!diplomka/standart.php?

info\_file\_name=".\$info\_file\_name."&zprava=Zadejte zmenu velikosti v procentech  
(napr. 130)");

exit;

}

else

{

header ("Location: ../cgi-bin/matweb.exe?

m1mfile=diplomka&info\_file\_name=".\$info\_file\_name."&akce=resize");

exit;

}

}

...

## **Příloha B**

### **Požadavky na hardware a software**

#### **SERVER**

HTTP Server s podporou PHP

MATLAB s podporou Image Processing Toolbox

MATLAB Web Server

Vývojová konfigurace

MATLAB 6.5 Release 13

MATLAB WEB SERVER TOOLBOX 1.2.2

MATLAB IMAGE PROCESSING TOOLBOX 3.2

HTTP SERVER APACHE

PHP Version 4.1.1

#### **KLIENT**

Jakýkoliv grafický internetový prohlížeč.

Testováno v internetových prohlížečích:

Internet Explorer 6.0

Mozilla 1.3b

Opera 7.01



## **Příloha C**

### **Obsah přiloženého CD-ROM.**

Součástí této zprávy je CD-ROM, na kterém jsou uloženy zdrojové kódy.

#### **Dokumentace**

V adresáři *|dokumentace* je tato zpráva ve formátu .PDF.

#### **Instalace**

V adresáři *|install* je instalace Java podpory, WWW serveru Apache i PHP.

#### **Zdrojové kódy**

V adresáři *|aplikace* jsou zdrojové kódy Matlab funkcí i výtvořené aplikace.